VŠB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Applied Mathematics

Image segmentation techniques in the HPC environment and their applications

Techniky segmentace obrazu v prostředí HPC a jejich aplikace

2016 Bc. Marek Pecha

VŠB - Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Applied Mathematics

Diploma Thesis Assignment

Student: **Bc. Marek Pecha**

Study Programme: N2647 Information and Communication Technology

Study Branch: 1103T031 Computational Mathematics

Title: Image segmentation techniques in the HPC environment and their applications.
Techniky segmentace obrazu v prostředí HPC a jejich aplikace.

The thesis language: English

Description:

An image decomposition into regions of interests, which are commonly called segments, is an integral part of modern techniques for classifying and processing a piece of information which is decoded from discrete image functions. Typically, the segments are interesting objects in an image-foreground scene, and they are apart from an insignificant background.

The goal of this diploma thesis is to study at least two different techniques for image segmentation and their massively parallel implementation for HPC platforms, and test them on real-world examples. One of the techniques will be based on a spectral clustering method.

The implemented techniques will be imported into PERMON Toolbox specific modules. PERMON Toolbox is developed by IT4Innovations National Supercomputing Center in Ostrava in cooperation with the Department of Applied Mathematics at VŠB-TUO.

References:

Y. SAAD, Iterative Methods for Sparse Linear Systems: Second Edition. Rev. ed. Philadelphia: Society for Industrial and Applied Mathematics, c2013, ISBN 978-0-898715-34-7.
J. R. Parker: Algorithms for Image Processing and Computer Vision, 2nd Edition, 2010. ISBN 978-0470643853

Extent and terms of a thesis are specified in directions for its elaboration that are opened to the public on the web sites of the faculty.

Supervisor: **Ing. Martin Čermák, Ph.D.**

Date of issue: 01.09.2015

Date of submission: 29.04.2016

doc. RNDr. Jiří Bouchala, Ph.D.
Head of Department

prof. RNDr. Václav Snášel, CSc.
Dean of Faculty

I declare that I have worked up this thesis by myself. I have referenced all the sources and publications I have used.

Ostrava, April 29, 2016 .....

Firstly, I would like to express my thanks to Dr. Martin Čermák, the supervisor and mentor of my thesis. He is a wonderful person, who believed in me, energized and encouraged me throughout the whole course of work on the thesis. Moreover, Renča Plouharová deserves my thanks, virtually a kiss and hug for helping me with English and delivering optimism to me. Furthermore, I would like to thank to Permon Team, my IT4Innovations family, for helping me in dark times. And of course, greatest thank belongs to my parents, who have been supporting me during the course of my university studies.

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project „IT4Innovations excellence in science - LQ1602“ and from the Large Infrastructures for Research, Experimental Development and Innovations project „IT4Innovations National Supercomputing Center – LM2015070“. This work was also supported by the internal student grant competition project SP2016/178 “PERMON toolbox development II” and by the Grant Agency of the Czech Republic (GACR) project no. 15-18274S.

The work is partially supported by Grant of SGS No. SP2016/108, VŠB-Technical University of Ostrava, Czech Republic.

Abstract

An image decomposition into regions of interests, which are commonly called segments, is an integral part of modern techniques for classifying and processing a piece of information which is decoded from discrete image functions. Typically, the segments are interesting objects in an image-foreground scene, and they are apart from an insignificant background.

The goal of this diploma thesis is to study at least two different techniques for image segmentation and their massively parallel implementation for HPC platforms, and test them on real-world examples. One of the techniques will be based on a spectral clustering method.

The implemented techniques will be imported into PERMON Toolbox specific modules. PERMON Toolbox is developed by IT4Innovations National Supercomputing Center in Ostrava in cooperation with the Department of Applied Mathematics at VŠB-TUO.

Keywords: spectral clustering, image segmentation, supercomputing, PERMON

Abstrakt

Rozdělení obrazu na oblasti zájmu (segmenty), které typicky představují významné prvky v popředí zachycené scény, oddělené od nevýznamného pozadí, je nedílnou součástí moderních technik pro klasifikaci a zpracování informací dekódovaných z diskrétních obrazových funkcí.

Cílem této diplomové práce je nastudování nejméně dvou různých technik pro segmentaci obrazu, z nichž jedna bude založena na metodě spektrálního shlukování, jejich masivně paralelní implementace pro HPC platformy a otestování na několika reálných úlohách.

Implementované techniky budou začleněny do specifických modulů PERMON Toolboxu, který je vyvíjen v Národním superpočítačovém centru IT4Innovations v Ostravě ve spolupráci s Katedrou aplikované matematiky VŠB-TUO.

Klíčová slova: spektrální shlukování, segmentace obrazu, supercomputing, PERMON

Contents

List of Figures

Acronyms

BMP Bitmap (file name extension)

CAS Czech Academy of Sciences

CEHIS Complete Extended Hierarchical Image Segmentation

CPU Central Processing Unit

CT Computerized Tomography

ECO Evolution-COnstructed

EHIS Extended Hierarchical Image Segmentation

GB GigaByte

GPU Graphics Processing Unit

JPEG Joint Photographic Experts Group (file name extension)

MB MegaBytes

MRM-RFC Method of Reference Materials for Reinforced Ferro Concrete

NP Nondeterministic Polynomial

PCS Problem Contex Sense

PDE Partial Differential Equation

PNG Portable Network Graphics (file name extension)

ROI Region of Interest

SMRM Simple Method of Reference Materials

SVM Support Vector Machines

TIFF Tagged Image File Format (file name extension)

Symbols

$\emptyset$	-	Empty set
$\mathbf{o}$	-	Null vector
$\mathbb{N}$	-	Set of all natural numbers
$\mathbb{N}^d$	-	Set of all d-dimensional natural vectors
$\mathbb{Z}$	-	Set of all integers
$\mathbb{Z}^d$	-	Set of all d-dimensional integer vectors
$\mathbb{C}$	-	Set of all complex numbers
$\mathbb{R}$	-	Set of all real numbers
$\mathbb{R}^d$	-	Set of all d-dimensional real vectors
$\mathbb{C}$	-	Set of all complex numbers
$\mathbb{C}^d$	-	Set of all d-dimensional complex vectors
$\mathbf{I}$	-	Identity matrix
$\mathbb{I}$	-	Index set
∇	-	Gradient
Δ	-	Laplace operator
f_{co}	-	Continuous image function
f_d	-	Discrete image function
f_{mask}	-	Image binary mask
f_{sd}	-	Semi-discrete image function
f^{ess}	-	Essential object
Ω	-	Image function domain
$\text{int}\,(\Omega)$	-	Interior of image function domain Ω
Ω_{roi}	-	Region of Interest domain
Ω^{roi}	-	Essential object domain
$\partial\Omega$	-	Set of image region boundaries
G_{im}	-	Set of all possible image functions
V_{im}	-	Pixel value set
$L_{\Omega_{roi}}$	-	Label set
$\mathbf{L}\,(f)$	-	Labelling operator
$\mathbf{C}\,(f)$	-	Object-characteristics operator
$\mathbf{U}_0$	-	Solution class
T_{ehis}	-	Extended Hierarchical Image Segmentation
T_{cehis}	-	Complete Extended Hierarchical Image Segmentation
T_{pehis}	-	Partial Extended Hierarchical Image Segmentation
$\mathcal{H}^d(E)$	-	d-dimensional Hausdorff measure of set E
$\mu(\Omega)$	-	Lebesgue measure

G	-	Image similarity graph
$V(G)$	-	Vertex set of graph G
$E(G)$	-	Edge set of graph G
$\mathbf{W}$	-	Graph adjacency matrix
$\mathbf{D}$	-	Graph degree matrix
$\mathbf{L}$	-	Laplace matrix
$\mathbf{L}_{rw}$	-	Laplace matrix of random walk
$\mathbf{L}_{sym}$	-	Symmetric version of $\mathbf{L}_{rw}$
$\mathbf{e}$	-	Eigenvector
λ	-	Eigenvalue

Chapter 1

Preface

> I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have "327"? No. I have sky, house, and trees.
>
> *Max Wertheimer*

Mathematics is a beautiful natural science; "her" appeal lies in its exactness and formalness. The word Mathematics comes from Greek μαΘημα *máthema* and it means "knowledge, study and learning," and *that's what it's all about*! This is true not only in the realm of science, but we learn throughout our whole life how to make good decisions, and that we cannot have everything at once; some things just take time and they will go wrong if we try to push them. As far as I am concerned, this thesis was an ordeal by fire and I am pretty sure that not only for me and I think not only for me. And those ones know. For the successful completion of this thesis, I had to learn to comunicate with people, to work in a team, and express complex thoughts. I would once again like to thank the people who believed in me and in my work, who energized, and encouraged me. Without them, the work would never come to light. Hence, I decided to write the text in a plural form.

Two years ago, I started to work in IT4Innovations National Supercomputing Center. I was a fresh graduated with bachelor's degree and my head has been full of ideas. I have met with wonderfull people in Permon Team and, at this time, thought about connecting human reality perception with the exact mathematics theory has turned out. Moreover, I wanted to create a useful practical application, which would be based on the theory. Therefore, I have developed an extensive cooperation with *University Hospital in Ostrava* and *Institute of Geonics of the CAS* and, precisely at this moment, "I" have been turned into "we".

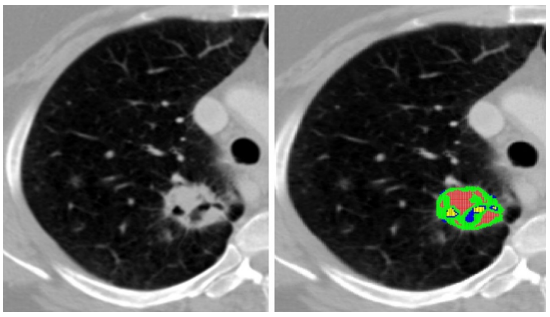
We may consider this text as the resume of a two-years work. Moreover, we have endeavoured to write a readable and unputdownable text, and we hope to that we have successfully managed it.

We may divide the text into 3 parts. In the first one, i.e. Chapter 2, we outline the image segmentation problems and introduce our quite new image segmentation theory, which is called *Extended Image Segmentation Theory*. It is based on measure and set theories and converts human reality perception into mathematics approaches in general. In Chapters 3 and 4, we present standard image clustering techniques, i.e. Lloyd type image clustering techniques and spectral clustering techniques, which are re-designed for our quite new image segmentation theoretical approaches. In the end, i.e. Chapter 5, we conclude the thesis with practical applications, because theory without practical application is useless ...

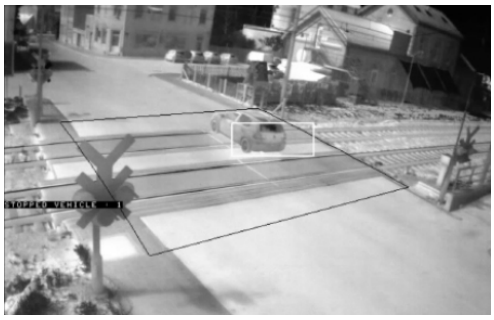
Chapter 2

Image Segmentation in General

In computer science - especially - in image processing, image segmentation plays an important role as a pre-processing technique for recognizing and evaluating visual inputs. Underneath the recognizing and evaluating processes, we can imagine, for example, detecting a carcinoma in CT scans of lungs or avoiding accidents at level crossings by evaluating critical situations in video streaming sequences, e.g. a trapped car or a person between barriers; see **Figures 2.1a, 2.1b**. Also, there are applications which we use in everyday life, e.g. in modern digital cameras, a face-detecting technology is integrated for an autofocus.



(a) **Example of an image segmentation in bio-medical applications:** a separation of a lung carcinoma tissiue from healthy parts of a lung. Referenced from [1, p. 11].



(b) **Example of an image segmentation in public-transport applications:** a detection of a vehicle at a level crossing [2].

Figure 2.1: Image segmentation in applications.

Despite the achievements in the segmentation of visual inputs, there are still difficulties which are not successfully resolved. One of the important difficulties is dissimilar interpretation of identical visual inputs.

In [3], we give the *Scientist-Workman-Hammer* example. We easily and non-mathematically explain the reason for dissimilar interpretations of the hammer from two different points of view. In the following theorem, we simplify ideas of the *Scientist-Workman-Hammer* example.

Theorem 1 – Scientist-Workman-Hammer. Two men, a scientist and workman, are looking at a hammer.

i) The scientist explores the hammer and proclaims it as an unneeded thing.

ii) The workman considers the hammer useful for buildings construction.

iii) The workman and scientist are both right.

Of course, the theorem is very simple but it is important to realize that human-thinking is complex and unique for every single individual. Hence, computer simulation of human perception of the reality is problematic to implement; it is not only because of its a computational complexity. During our life-time, we make memories, learn to react to visual impulses, recognize objects, and categorize them. We can quickly react to changes in our surroundings. For the most of us, creativity is an essential part of our abilities, and this is an one of the causes why the human brain can process non-complete information. Obviously, a human brain is a complex system of the neuronal tissue, synapses, etc. In a one paragraph, it is not possible to describe it; and it is not the goal of this thesis. For more information of *how the brain solves visual object recognition*, see [4].

At the moment, someone would argue: "*But there is high-accuracy genetic algorithm for image recognition. Why is not sufficient for so-called automatic image segmentation and automatic objects recognition?*"

Yes, it is true; the algorithm is really exists, and it is called ECO[a] [5]. Moreover, there are other research projects which are focused on objects identification in image scene, e.g. Wolfram Image Identification Project (often called *WolfAI*) [6] or the Googles Self-Driving Car Project [7]: an autonomous car detects, evaluates situations on the roads, generates potential reactions and processes the best one of them at the time.

Nevertheless, the Google Self-Driving Car is application-specific and therefore it is inappropriate for so-called automatic segmentation of images or object recognition processes. On the other side, the WolfAI's algorithm could be a good candidate but it is tested only on a small dataset (about 10,000 objects), and in some cases, its answers are absolutely incorrect, see **Figure 2.2**. However, it is not design-fault of the algorithm, which is based on deep neural networks [8], but there is an information-mining problem and the problem of *how to digital storing amounts of image transformations without losing the important properties.*

[a] ECO stands for Evolution-COnstructed.

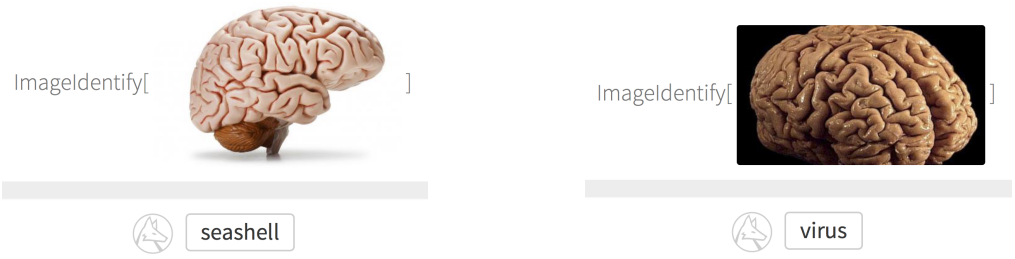


Figure 2.2: WolfAI's ImageIdentify superfunction testing.
Testing of the Wolfram's image identification project on images of human brains. The testing images are referenced to [9] *and* [10].

Thus, the one of the mentioned algorithms still remains - the ECO. In our opinion, it would be a small step for developing advanced artificial intelligence and its human-like recognition of objects in a visual scene. The ECO is a self-tuned, genetic pattern recognition algorithm with sub-region selection features. It means, the algorithm decides which objects, or parts of the objects, could be important in a visual scene, and then it extracts their main features (see **Figure 2.3**). Moreover, the algorithm is 100% successful for image identification for the Caltech Image Dataset [11]. However, are the achieved results sufficient for automatic processes on images?

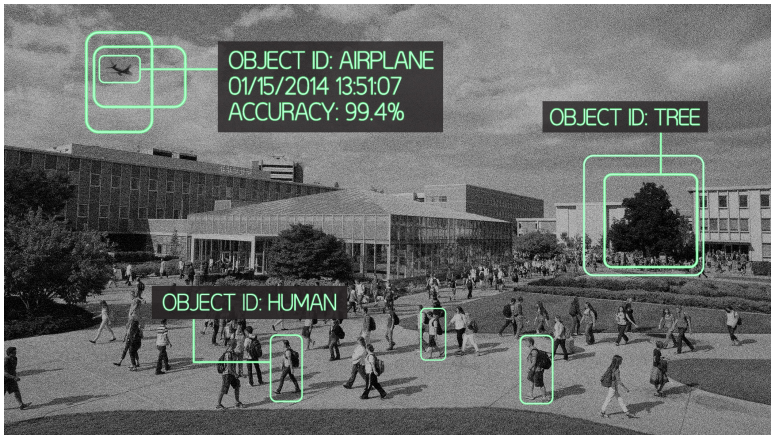


Figure 2.3: ECO algorithm sub-region selection feature.
The algorithm decides which objects could be important in a visual scene [12].

Even if the ECO's image segmentation and recognition techniques are quite advanced and highly accurate, we think that so-called automatic image segmentation is not possible at the moment.[a] The first problem is mentioned at the beginning of this chapter - in the *Scientist-Workman-Hammer* problem. We can consider it as the non-mathematical paraphrase of unfulfilling the assumptions of the *well-posed problem* definition.

[a] However in our opinion, it will be accomplished (although with few restrictions) in the future.

It is obvious, well-posed problem [13] is defined as follows: *the solution exists, it has to be unique and the solution's behavior is continuously changed with initial conditions*. Further, let's consider that we have two perfect classifiers. The first of them classifies the reality from the workman's point of view and the second one percepts the visual inputs from the scientist's point of view. We set up a task for a recognition and marking-up suitable objects in a hammer-image, and after that we assign the task to the classifiers. For example, the first classifier would return `tool` as the recognition answer and marks up the hammer object. The second one would recognize the head and the handle of the hammer and marks them up. As we know, both solutions are right - but are not unique.
Even if the image recognition and segmentation processes are often defined as well-posed, from the mathematical point of view it is not true. To avoid this problem, we introduce the solutions in a specific form (we call it *solution-class*), in Section 2.2. However, we assume that image recognition and segmentation processes are not well-posed.

If we do not take into account the image segmentation well-posedness, and we consider the achieved solution to be unique, in specific sense, then another problem arises.
In Section 2.3, we introduce Kleinberg's Impossibility theorem too. Major idea of this theorem is: *there is not a general function for the general image segmentation and recognition problems*. Therefore, a classifier has to be adaptable for the image segmentation and recognition tasks, e.g. by an automatic reimplementation of the source code.

By automatic reimplementation of the source code we mean: *based on inputs, the program is capable of self-reimplementing its structure or parts of its source code, not only changing variable values. This procedure would correspond to natural evolution of not only the human brain, i.e. the brain creates new neurons and connects them by synapses, controls, and changes the chemical environment in the surroundings.*[a] Above that, nowadays, it is not possible to teach computers how to think creatively.

Hence, we define automatized image segmentation and object recognition - for specific applications - as the *semi-automatic* process.

[a] In machine learning and cognitive science, there were experiments with automatic determination of an optimal count of neural network layers and an optimal count of neurons in each layer [14], [15]. Essentially, it is pruning of oversized neural networks and in practice this approach is not commonly used.

2.1 Mathematical representations of images

To efficiently handle principles of image segmentation (or an image processing techniques which are used in image segmentation processes); first, we have to understand mathematical representations of images and which adequate properties we treat for them.
For the purposes of this text, we define two groups of digital images: *synthetic images*, which are created in drawing programs from scratch, and *real images*, e.g. photos.

First, we describe the second group - the *real images*. The visual environment around us is naturally analog. By analog, we mean that signals - visible light waves - exist on a continuous space and time domain. But for the computer image segmentation (processing) techniques, the real-world signals must be converted into a computer-readable format, the non-continuous digital representation. It means an analog signal must be converted into the discrete space and time domain (A/D conversion). For the representation of images, we consider only the space domain at the time-point, e.g. at the time-point, when we pulled a trigger of a camera. For more information about image digitalizing process (or subprocesses such as *sampling* and *quantization*, etc.), see, e.g. the following books: *Handbook of image and video processing* [16, pp. 5-13] or *Digital picture processing* [17, pp. 106-114].

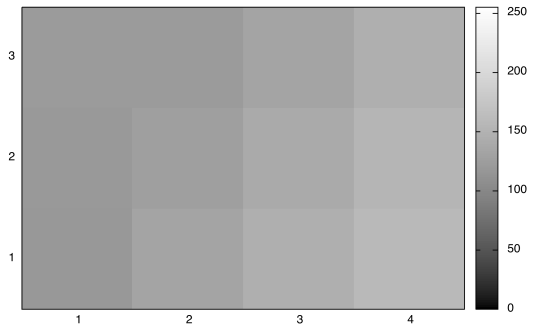


Figure 2.4: Matrix representation of an image.
Matrix representation of a synthetic image which has resolution 4×3 *px.*

After the digitalization process of an analog visual signal, the values of the discrete representation of this signal are digitally stored into a matrix (see **Figure 2.4**). Its cells are called pixels - *picture elements* - and a row and column count determines the resolution of an image.
For grayscale images, the values of pixels are typically small integers, e.g. for an 8-bit depth - one channel - image[1], the values of pixels belong to a set $V_{im} := \{0,\ 1,\ 2,\ \ldots,\ 255\}$. In case of colour images, the pixels are typically represented by three components-vectors[2], in which each component has values from V_{im}.
Thereafter, the matrix of pixels is compressed into an image file[3] by a method, which is specified for the image format, e.g. *JPEG* uses Huffman coding [18] for a lossless [19] compression, and decompression of the matrix of pixels; whereas in a modern *JPEG*, known as *JPEG 2000*, the matrix of pixels is lossy [20] or lossless compressed, and decompressed by the wavelet transforms techniques [21]. Of course, there are approaches of an uncompressed storage of the pixel matrices. The typical file formats are, e.g. *BMP* or *RAW*.
Identical properties are also applied for the pixels and pixel-matrices of synthetic images; hence, unless otherwise stated, by the term *image* we mean a real or synthetic image.

[1] It is a common grayscale image.
[2] Hence, a 24bit depth colour True Color image we often call 3 channels image.
[3] A major file formats are e.g. *PNG*, *JPEG*, *TIFF*.

In this text, we mention three definitions of images - or better *image functions* - for an image segmentation process or, generally, for image processing techniques.

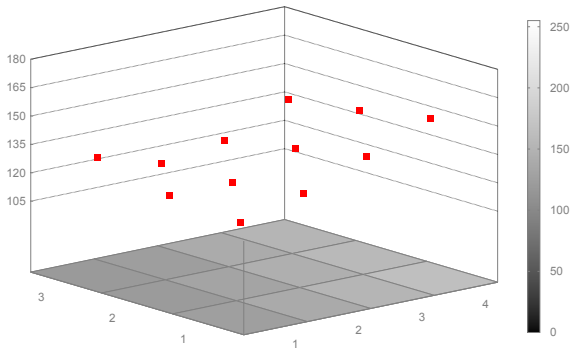


Figure 2.5: Discrete image function.
*Representation of an image in **Figure 2.4** as a discrete image function.*

The first of them is straightforward. We consider a digital image as a discrete function as follows: we stepwise assign appropriate values to discrete points by following approach. Let $p_1,\ p_2,\ p_3,\ \ldots,\ p_n \in \mathbb{N}^2$ be discrete points of a discrete image function; appropriate coordinates of the points are stepwise $(1,1),\ (2,1),\ \ldots,\ (c,1),\ \ldots,\ (r,1),\ (r,2),\ \ldots,\ (r,c)$, where c is a column count of the input digital image and r is its row count. We denote a discrete image function

$$f_d : \mathbb{N}^2 \to V_{im} \tag{2.1}$$

for common grayscale images, see **Figure 2.5**, or

$$f_d : \mathbb{N}^2 \to V_{im}^3 \tag{2.2}$$

for typical colour images, where $V_{im}^3 := \{(r,g,b) : r,g,b \in V_{im}\}$.

For two following definition, let $\widetilde{\mathbb{R}}^2 \subset \mathbb{R}^2$ as follows:

$$\widetilde{\mathbb{R}}^2 := \{(x,y) : x,y \in \langle 1, +\infty)\}.$$

In the second definition, we extend the domain of the discrete function f_d as follows: we add real values between natural indices belonging to an x-axis and a y-axis, and for the square element

$$\langle x, x+1) \times \langle y, y+1),$$

we define a constant range of values which is equaled $f(x,y)$. This function will be called a *semi-discrete* image function and it will be denoted

$$f_{sd} : \widetilde{\mathbb{R}}^2 \to V_{im} \tag{2.3}$$

for common grayscale images, see **Figure 2.6**, or

$$f_{sd} : \widetilde{\mathbb{R}}^2 \to V_{im}^3 \tag{2.4}$$

for typical colour images.

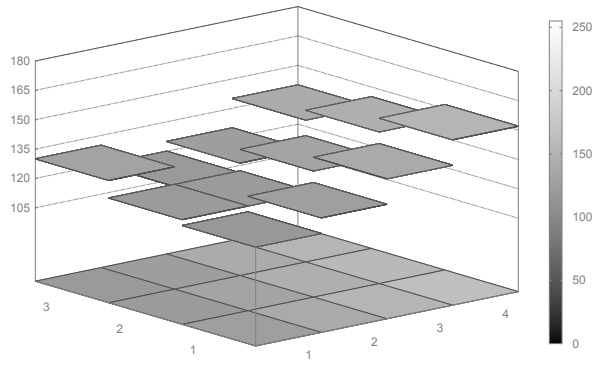


Figure 2.6: *Semi-discrete* image function.
*Representation of an image in **Figure 2.4** as a semi-discrete image function.*

For the third – remaining - mathematical representation of images, we use the following real image, in the **Figure 2.7**. It is a non-mainstream photo of Lena Söderberg - November72 playmate from Sweden.

Lenna Story *Lena Söderberg appeared as a Playmate in the November 1972 issue of Playboy magazine under the name Lenna Sjööblom. Her centerfold was photographed by Dwight Hooker. A section of her centerfold (known as Lenna) is often used to test algorithms in digital image processing. She was a guest at the 50th annual Conference of the Society for Imaging Science and Technology in 1997 where she gave a presentation about herself. Because of the ubiquity of her Playboy photo scan, she has been called the "first lady of the internet."* This text is referenced from wikipedia.org.

Often, for image processing mathematical models, especially, in general *PDE approaches* or *variational PDE* models, we require continuous image functions with appropriate properties. For example, we treat them as general L^2 functions in the wavelet analysis methods [23, p. 239] or in the active contour method, which is based on minimization of *Mumford and Shah functional*, we require for an input function $g(x, y)$ to be $L^\infty(\mathbb{R}^2)$ and for a suitable solution $u(x, y)$ we treat $u(x, y) \in W^{1,2}(\mathbb{R}^2)$ [24, pp. 1-3].

For the last image function definition, we introduce a set $\widetilde{\mathbb{R}}_{im}^3 \subset \mathbb{R}^3$ as follows:

$$\widetilde{\mathbb{R}}_{im}^3 := \{(r, g, b) : r, g, b \in \langle 0, 255 \rangle\}.$$

Figure 2.7: Lena Söderberg.
The picture is referenced from [22].

The construction of continuous image function can be easily done by continuous interpolation of the f_d sample points, linear or spline interpolations are typically used. We denote a continous image function such that:

$$f_{co} : \widetilde{\mathbb{R}}^2 \to \langle 0, 255 \rangle \tag{2.5}$$

for common grayscale images, see **Figure 2.8**, or

$$f_{co} : \widetilde{\mathbb{R}}^2 \to \widetilde{\mathbb{R}}^3_{im} \tag{2.6}$$

for typical colour images.

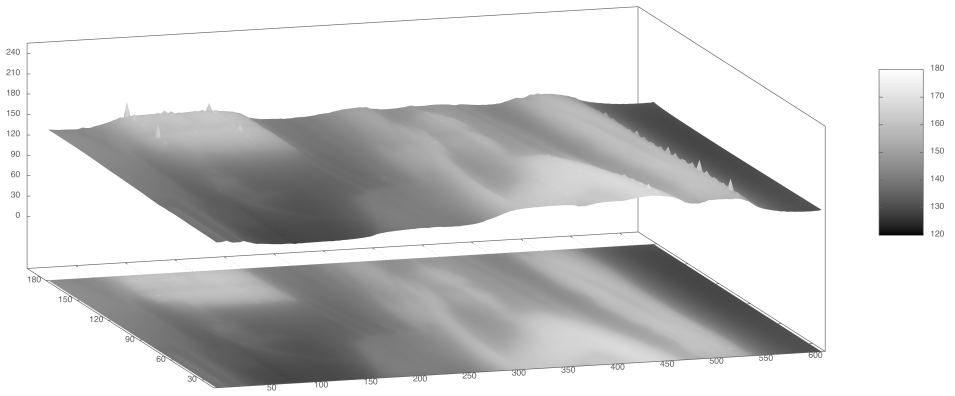


Figure 2.8: Continuous image function.
*Representation of a picture in **Figure 2.7** as a continuous image function.*

Thus, unless otherwise stated, we define an image function $f(x, y)$ as the continuous image function, i.e.

$$f(x, y) := f_{co}(x, y). \tag{2.7}$$

2.2 Fundamental notation, definitions and theorems

Let $g(x, y) \in G_{im}$ be an image function, where G_{im} is a set of all possible image functions. As we know, image segmentation processes are not well-posed (in general), there is not only one solution u but a set $U := \{u_1, u_2, \ldots, u_m\}$ of right solutions. Hence, we have to determine convenient properties of image segmentation procedures, their mathematical models and their correct mathematical interpretations.
For example, to avoid obstacles with non-unique solutions, we redefine the solution u as a *solution-class*, which we consider unique. Then, we can easily establish a theory of "semi-approaches" for a mathematical modeling of image segmentation procedures and well-posedness of the procedures (in "semi-approach" sense).

Let $\Omega \subset \widetilde{\mathbb{R}}^2$ be an image function area. Therefore, it is suitable to redefine an image function $f(x, y)$ as the $f : \Omega \to \langle 0, 255 \rangle$ or in case of a colour image as the $f : \Omega \to \widetilde{\mathbb{R}}^3_{im}$.

Definition 1 – Region of Interest. The *Region of Interest* $\Omega_{roi} \subset \Omega$ is a part of an image function area, which is selected by a segmentator/classifier – a program or human user - to effectivety utilize a CPU/GPU computational time and its reduction.[a]
Ω_{roi} is typically part of an image foreground and $\Omega \setminus \Omega_{roi}$ is interpreted as the background of an image, see **Figure 2.9**.

[a] In general, the *Region of Interest* is used for utilization of a running computation in any computational unit and any computational environment.

In scientific publications, the *Region of Interest* is often denoted as the acronym *ROI*. For the purposes of this text, the *ROI* definition is sufficient, therefore we refer the book *The Art and Science of Digital Compositing* [25, p. 340] for further information.

(a) **Tatra T603** (b) **Region of Interest**

Figure 2.9: Region of Interest: Tatra T603.
The ROI of Tatra T603 image [26] *is created by selecting proper external Canny edges* [27].

For extracting the *Region of Interest* from an image function area Ω, a *binary mask* is used in an image processing field. In general, a binary mask corresponds to *a characteristic function of a set*, see **Definition 2**. In mathematics, *a characteristic function* of a set is commonly known in the variational methods (not only) for the definition of the *Lebesgue integral*.

Definition 2 – Binary mask. Let Ω be an image area, Ω_{roi} be the *Region of Interest*, and f be an image function. *An image binary mask* $f_{mask} : \Omega \to \{0, 1\}$ determines points $p_1,\ p_2,\ \ldots,\ p_n \in \Omega$ which belong to Ω_{roi}, see **Figure 2.10a**, and it is defined as the *characteristic function* of Ω_{roi}

$$f_{mask}(p_j) := \begin{cases} 1 & \text{if a point } p_j \in \Omega_{roi}, \\ 0 & \text{if a point } p_j \notin \Omega_{roi}. \end{cases} \tag{2.8}$$

The binary mask is applied on an image by a pointwise product of an image function $f(x, y)$ and the appropriate *binary mask* $f_{mask}(x, y)$, see **Figure 2.10b**.

$$f_{roi} := f_{mask} \cdot f. \tag{2.9}$$

(a) ROI binary mask (b) **Figure 2.9a** after binary mask application

Figure 2.10: Binary mask and its application.

2.2.1 Extended Image Segmentation Theory

Before establishing the definition of a specific form of an image segmentation solution, which we call a *solution-class*, it is still necessary to introduce several definitions from set and measure theories, i.e., *Diameter of a set*, which is referenced from [28], δ-cover of a set, which is referenced from [28] too, and *Haussdorf measure* [29, pp. 81 - 100].

Definition 3 – Diameter of a set. Let (X, ρ) be a metric space. For an arbitrary set $M \subseteq X$, **diam** M is denoted as the diameter of M such that

$$\textbf{diam } M := \sup \{\rho(x, y) : x, y \in M\} \tag{2.10}$$

Further, we define

$$\textbf{diam } \emptyset := 0, \tag{2.11}$$

and $(\forall N \subset X)\,(|N| = 1)$ we define

$$\textbf{diam } N := 0. \tag{2.12}$$

Definition 4 – δ-cover of a set. Let (X, ρ) be a metric space and $\delta > 0$. δ-cover of $E \subset X$ is a countable collection of subset $\{M_1,\ M_2,\ \ldots,\ M_m\} \subset X$ such that

$$(m \in \mathbb{N})\,(E \subseteq \cup_{i=1}^{m} M_i \wedge 0 < \textbf{diam } M_i < \delta)\,. \tag{2.13}$$

Definition 5 – Haussdorf measure. Let E be a subset of $\mathbb{R}^n$ and $\delta > 0$. For any $s \geq 0$, we define

$$\mathcal{H}^s_\delta(E) := \inf \left\{ \sum_{i=1}^{m} (\textbf{diam } M_i)^s : \{M_1,\ M_2,\ \ldots,\ M_m\} \text{ is } \delta\text{-cover of } E \right\}. \tag{2.14}$$

For $\delta \to 0$, the limit

$$\lim_{\delta \to 0} \mathcal{H}^s_\delta(E) \tag{2.15}$$

is called *s-dimensional Hausdorff measure* of E and is denoted as $\mathcal{H}^s(E)$. Moreover,

$$\mathcal{H}^s(E) = \lim_{\delta \to 0} \mathcal{H}^s_\delta(E) = \sup_{\delta > 0} \mathcal{H}^s_\delta(E). \tag{2.16}$$

Definition 6 – Labels-set. Let $\Omega_l \subseteq \Omega_{roi}$ be a compact part of an image-area Ω and $f(x, y)$ be an image function, which is defined on Ω. $f_{|\Omega_l}$ is called an object if a clear and a reasonable naming of $f_{|\Omega_l}$ exists (in a problem-context sense). We will call the namings *labels*, a set $\{l_1,\ l_2,\ \ldots,\ l_k\}$ of all *labels* (of the problem) is called a *label-set* and we denote it

$$\mathbb{L}_{\Omega_{roi}} \text{ (or } \mathbb{L}_{\Omega}). \tag{2.17}$$

Further, we define a *labeling-operator* $\mathbf{L} : G_{im} \to \mathbb{L}_{\Omega_{roi}}$ such that

$$\mathbf{L}\left(f_{|\Omega_j}\right) := \begin{cases} \text{unique } l_i \in \mathbb{L}_{\Omega_{roi}} & \text{if } f_{|\Omega_j} \text{ is an object,} \\ \emptyset & \text{otherwise.} \end{cases} \tag{2.18}$$

Definition 7 – Solution-class. Let

- $i, j, k, m, n, p, q, r, s, t, v$ be auxiliary variables,
- $\Omega \subset \widetilde{\mathbb{R}}^2$ be an image-area,
- $f(x, y)$ be an image-function, which is defined on Ω,
- $\Omega_{roi} \subseteq \Omega$ be the *Region of Interest*,
- $\mathbb{L}_{\Omega_{roi}}$ be a problem-specified *label-set*,
- $\mathbf{L} : G_{im} \to \mathbb{L}_{\Omega_{roi}}$ be a *labeling-operator*,
- $\mathcal{H}^1(\Omega)$ be the *1-dimensional Hausdorff measure*,
- $\mathcal{H}^0(\Omega)$ be the *0-dimensional Hausdorff measure*,
- $\mu(\Omega)$ be the *Lebesgue measure*,
- and a reasonable large $\varepsilon > 0$.

$f_{|\Omega_i}$, where $\Omega_i \subset \Omega_{roi}$, is *an essential object* if for all $\Omega_j \subset \Omega_i : \mathbf{L}(f_{|\Omega_j}) = \emptyset$. We denote *an essential object* as the function $f^{ess}_{|\Omega_i}$, which is defined on *an essential part* Ω^{ess}_i of an image area Ω_{roi}.
Let $f_{|\Omega_m}$, where $\Omega_m \subset \Omega_{roi}$, be an object, which is not *essential*, and $\{\Omega^m_1,\ \Omega^m_2,\ \ldots,\ \Omega^m_n\} \subset \Omega_m$ is a set of *essential* or non-*essential* object areas; we call $f_{|\Omega_k}$, where $\Omega_k := \Omega_m \setminus \{\Omega^m_1,\ \Omega^m_2,\ \ldots,\ \Omega^m_n\}$, *a semi-essential object* if

$$\Omega_k \neq \emptyset \ \wedge\ \mathbf{L}(f_{|\Omega_k}) = \emptyset. \tag{2.19}$$

We denote a *semi-essential object* as the function $\widetilde{f^{ess}_{|\Omega_k}}$, which is defined on *a semi-essential part* $\widetilde{\Omega^{ess}_k}$ of an image area Ω_{roi}.

Therefore, we define a *solution-class* as a set of objects and essential objects such that

- $f_{|\Omega_p}$, where $\Omega_p \subset \Omega_{roi}$, is an object and $f_{|\Omega_q^p}$, such that $\Omega_q^p \subset \Omega_p$, $f_{|\Omega_r^p}$, such that $\Omega_r^p \subset \Omega_p$, are objects or *essential objects* and $\Omega_q^p \not\subset \Omega_r^p \ \wedge \ \Omega_r^p \not\subset \Omega_q^p$ so

$$\mu(\Omega_q^p \cap \Omega_r^p) = 0, \tag{2.20}$$

$$\mathcal{H}^1(\Omega_q^p \cap \Omega_r^p) \begin{cases} > 0 & \text{if } \partial\Omega_q^p \cap \partial\Omega_r^p \neq \emptyset \text{ and} \\ & \partial\Omega_q^p \cap \partial\Omega_r^p \text{ is Hausdorff 1-measurable,} \\ = 0 & \text{otherwise,} \end{cases} \tag{2.21}$$

$$(\mathcal{H}^1(\Omega_q^p \cap \Omega_r^p) = 0 \ \wedge \ \Omega_q^p \cap \Omega_r^p \neq \emptyset) \ \Leftrightarrow \ 1 \leq \mathcal{H}^0(\Omega_q^p \cap \Omega_r^p) < +\infty. \tag{2.22}$$

- $\widetilde{f^{ess}_{|\Omega_s}}$, such that $\widetilde{\Omega_s^{ess}} \subset \Omega_p$, exists so

$$\Gamma_1 := \partial\Omega_q^p \ \cap \ \partial\widetilde{\Omega_s^{ess}}, \ \mathcal{H}^1(\Omega_q^p \cap \widetilde{\Omega_s^{ess}}) = \mathcal{H}^1(\Gamma_1), \tag{2.23}$$

and

$$\Gamma_2 := \partial\Omega_r^p \ \cap \ \partial\widetilde{\Omega_s^{ess}}, \ \mathcal{H}^1(\Omega_r^p \cap \widetilde{\Omega_s^{ess}}) = \mathcal{H}^1(\Gamma_2). \tag{2.24}$$

- Another requirement for the objects and the essential objects is their size. Let $f_{|\Omega_t}$, where $\Omega_t \subset \Omega_{roi}$, be an object or *an essential object* that we consider $\mu(\Omega_t) \geq \varepsilon$.

Further, we stepwise denote all *objects* and *essential objects*, in a problem-context sense (PCS), as the $u_1, \ u_2, \ \ldots, \ u_v$ therefore we denote a *solution-class* $\mathbf{U}_0 := \{u_1, \ u_2, \ \ldots, \ u_v\}$.

Unless otherwise stated, by objects we mean the `objects` or *essential objects*. Further, we denote a set of object area boundaries $\partial\Omega_1, \ \partial\Omega_2, \ \ldots, \ \partial\Omega_n$ as $\partial\Omega$.

> Moreover, if we establish an appropriate factor operator for the decomposition of an image function area Ω into objects, the definition will be quite similar to the definition of continuous modular decomposition [30].
> However, a proper proof is beyond the scope of this work.

As we say at the beginning of this chapter: " *. . . for the most of us creativity is essential part of our abilities and this is an one of the causes why the human brain can process non-complete information. . . .* "
Let's imagine the following situation: in Båstnäs scrap yard, there are various types of vehicle wrecks, see **Figure 2.11**.

Further, we consider the sequence of Båstnäs scrap yard pictures as visual inputs, i.e. the input data for an image segmentator/classifier and we set up the image segmentation task on the data: *return and mark up the parts of an image area* Ω *which contains cars.*
Intuitively, a human image segmentator/classifier marks up parts of the pictures, in which the car wrecks are. But how to specify for an artificial segmentator/classifier, i.e. computer program,

Figure 2.11: Båstnäs scrap yard.
The pictures are referenced from [31].

what still is a car? For a car wreck, a few of general characterictics of a car are missed such as the characterictic of wheels, an engine, a steering wheel, etc.

Båstnäs scrap yard Story *Deep in the forest of the village of Båstnäs in Värmland, thousands of car wrecks from the 1930s to the 1970s form a junkyard. It's a unique place, owned by two brothers, Rune and Tore Ivansson, who used to run it as a commercial business for decades. Today, the site is the most spectacular graveyard for vintage cars in Sweden and possibly Europe. It has become a magnet for creative people and automobile enthusiasts.* The text is borrowed from a website steneby.se.

Let's try to imagine a following clearer example: *Let's have the application for tracking cars and measurement of their speed on a highway* [32]. *The input data is a sequence of pictures e.g. from a webcam, which is placed on a banister of a bridge.*
Depending on the camera location and a highway surrounding, the segmentator/classifier would not receive complete visual information about the detected car, e.g. a few parts of the car would be hidden behind other objects such as another car, a part of a tree, etc. A human classifier considers it as the same car, but for the program it is not same because few of object specification are missing.
To avoid presented obstacle, we establish an operator for characterizing objects, which we introduce in following definition. We note: this operator would be a part of a *labeling-operator.*

Definition 8 – Characteristic of objects. Let $\mathbf{L}\left(f_{|\Omega_i}\right) = l_k$ where $\Omega_i \subseteq \Omega_{roi}$. Further, let $C := \{c_1,\ c_2,\ \ldots,\ c_n\}$ be a set of reasonable characterictics which is unique (depending on a problem context) for every existed object or a set of objects with similar properties.
The operator $\mathbf{C} : f \to C$ is called *an object-characteristics operator*. Let $\delta > 0$ be resonable large and $\Omega_e \subset \Omega_i : \mu\left(\Omega_e\right) < \delta$ so

$$\mathbf{C}\left(f_{|\Omega_i}\right) \overset{\text{PCS}}{\approx} \mathbf{C}\left(f_{|\Omega_i \setminus \Omega_e}\right) \Rightarrow \mathbf{L}\left(f_{|\Omega_i \setminus \Omega_e}\right) := l_k. \tag{2.25}$$

For artificial image segmentators/classifiers, it is important to specify topology of objects, in a visual scene, for effective utilization of computational unit process-time. It means, to specify every possible object positions relatived to other objects and their positions to avoid solving inappropriate object locations, e.g. one point belongs to two different essential objects.
In the following theorem, we present this idea - `every possible topology among objects`.

Theorem 2 – Topology of objects. Let

- $i, j \in \{1,\ 2,\ \ldots,\ |U_0|\}$ and $i \neq j$,
- $f_{|\Omega_i} \in \mathbf{U}_0$,
- $f_{|\Omega_j} \in \mathbf{U}_0$

then

$$\Omega_i \subset \Omega_j \veebar\ \Omega_i \not\subset \Omega_j. \tag{2.26}$$

Proof 1 Let i, j be arbitrary but fixed, integers such that $i, j \in \{1,\ 2,\ \ldots,\ |U_0|\}$ and $i \neq j$, therefore

$$\left(f_{|\Omega_i} \in \mathbf{U}_0 \ \wedge\ f_{|\Omega_j} \in \mathbf{U}_0\right) \Rightarrow \left(\mathbf{L}(f_{|\Omega_i}) \neq \emptyset \ \wedge\ \mathbf{L}(f_{|\Omega_j}) \neq \emptyset\right) \Rightarrow (\Omega_i \subseteq \Omega_{roi}) \wedge (\Omega_j \subseteq \Omega_{roi}) \Rightarrow$$
$$\Rightarrow (\Omega_i \text{ is essential } \veebar\ \Omega_i \text{ is not essential }) \wedge (\Omega_j \text{ is essential } \veebar\ \Omega_j \text{ is not essential }) \Rightarrow$$

$$\Rightarrow \overbrace{(\Omega_i \text{ is essential } \wedge\ \Omega_j \text{ is essential})}^{A} \veebar \overbrace{(\Omega_i \text{ is essential } \wedge\ \Omega_j \text{ is not essential})}^{B} \veebar$$
$$\veebar \underbrace{(\Omega_i \text{ is not essential } \wedge\ \Omega_j \text{ is essential})}_{C} \veebar \underbrace{(\Omega_i \text{ is not essential } \wedge\ \Omega_j \text{ is not essential})}_{D} \Rightarrow$$

Proof of a term A

$$(\Omega_i \text{ is essential } \wedge\ \Omega_j \text{ is essential}) \Rightarrow [\neg(\Omega_i \subset \Omega_j) \ \wedge\ \neg(\Omega_j \subset \Omega_i)] \Rightarrow$$

$$\Rightarrow [\neg(\Omega_i \subset \Omega_j) \ \wedge \ \neg(\Omega_j \subset \Omega_i)] \ \Rightarrow \ [(\exists k \in \Omega_i : k \notin \Omega_j) \ \wedge \ (\exists l \in \Omega_j : l \notin \Omega_i)] \ \Rightarrow$$

$$\Rightarrow \text{Let } K := \{k_1,\ k_2,\ \ldots,\ k_m\} \text{ such that } K \subset \Omega_i \ \wedge \ K \not\subset \Omega_j \ \text{ and}$$

$$Q := \{q_1,\ q_2,\ \ldots,\ q_n\} \text{ such that } Q \subset \Omega_j \ \wedge \ Q \not\subset \Omega_i \ \Rightarrow \ \ P := (\Omega_i \cup \Omega_j) \ \setminus \ (K \cup Q) \ \Rightarrow$$

$$\overset{\text{Definition 8}}{\Longrightarrow} \mathbf{L}(f_{|P}) = \emptyset \ \Rightarrow \ P \text{ is semi-essential } \overset{\Omega_i,\,\Omega_j \text{ are compact and essential}}{\Longrightarrow}$$

$$\Rightarrow \left.\begin{cases} \Gamma_1 := \partial\Omega_i \cap \partial P \ \Rightarrow \ \mathcal{H}^1(\Omega_i \cap P) = \mathcal{H}^1(\Gamma_1) \\ \Gamma_2 := \partial\Omega_j \cap \partial P \ \Rightarrow \ \mathcal{H}^1(\Omega_j \cap P) = \mathcal{H}^1(\Gamma_2) \end{cases}\right\} \text{ and } P \subset \Omega_i \ \wedge \ P \subset \Omega_j \ \Rightarrow$$

$$\Rightarrow \ (\Gamma_1 = \Gamma_2 \ \Rightarrow \ \Gamma := \Gamma_1) \ \Rightarrow \ \underbrace{\partial\Omega_i \cap \partial\Omega_j = \Gamma \neq \emptyset}_{A1} \ \veebar \ \overbrace{\partial\Omega_i \cap \partial\Omega_j = \emptyset}^{A2} \ \Rightarrow$$

$$\Rightarrow \left\{\begin{array}{ll} (A1) & 0 < \mathcal{H}^1(\Omega_i \cap \Omega_j) < +\infty \Rightarrow \mu(\Omega_i \cap \Omega_j) = 0 \Rightarrow \Omega_i \not\subset \Omega_j \\ (A2) & \mu\left(\underbrace{\Omega_i \cap \Omega_j}_{=\,\emptyset}\right) = 0 \ \Rightarrow \ \Omega_i \not\subset \Omega_j \end{array}\right\} \ \Rightarrow \ \Omega_i \not\subset \Omega_j \ \square$$

Proof of a term B

$$(\Omega_i \text{ is essential } \wedge \ \Omega_j \text{ is not essential}) \ \Rightarrow \ \Omega_i \subset \Omega_j \ \veebar \ \underbrace{[\neg(\Omega_i \subset \Omega_j) \ \wedge \ \neg(\Omega_j \subset \Omega_i)]}_{\Longrightarrow \Omega_i \not\subset \Omega_j} \ \Rightarrow$$

$$\Rightarrow \Omega_i \subset \Omega_j \ \veebar \ \Omega_i \not\subset \Omega_j \ \square$$

Proof of a term C

$$(\Omega_i \text{ is not essential } \wedge \ \Omega_j \text{ is essential}) \ \Rightarrow \ \Omega_j \subset \Omega_i \ \veebar \ \underbrace{[\neg(\Omega_i \subset \Omega_j) \ \wedge \ \neg(\Omega_j \subset \Omega_i)]}_{\Longrightarrow \Omega_i \not\subset \Omega_j} \ \Rightarrow$$

$$\Rightarrow \ \Omega_j \subset \Omega_i \ \veebar \ \Omega_i \not\subset \Omega_j \ \square$$

Proof of a term D

$$(\Omega_i \text{ is not essential } \wedge \ \Omega_j \text{ is not essential}) \ \Rightarrow \ \Omega_i \subset \Omega_j \ \veebar \ \Omega_j \subset \Omega_i \ \veebar$$

$$\veebar \ \underbrace{[\neg(\Omega_i \subset \Omega_j) \ \wedge \ \neg(\Omega_j \subset \Omega_i)]}_{\Longrightarrow \Omega_i \not\subset \Omega_j} \ \Rightarrow \ \Omega_i \subset \Omega_j \ \veebar \ \Omega_j \subset \Omega_i \ \veebar \ \Omega_i \not\subset \Omega_j \ \square$$

We know that i, j are arbitrary but fixed, integers and $i \neq j$, therefore we clearly finish this proof

$$\Rightarrow \ A \ \veebar \ B \ \veebar \ C \ \veebar D \ \Rightarrow \ \Omega_i \subset \Omega_j \veebar \ \Omega_i \not\subset \Omega_j. \ \blacksquare$$

Corollary 1 From **Definitions 6 - 8**, **Theorem 2**, and its proof above, directly implies that the boundaries of objects are semi-essential. Moreover, an indirect consequence of **Theorem 2** is that: an arbitrary point p as follows,

$$p \in \Omega_{roi} \ \wedge \ p \notin \partial\Omega_{roi},$$

belongs to one essential object at most.

The last consequence, from the corollary above, allows us, quite clearly, to introduce a hierarchical decomposition of an image scene. It corresponds to human-like perception of visual inputs; first, we notice important objects in a visual scene, e.g. a car or a beautiful woman on a street, thereafter we observe their details such as a car brand, or the green colour of the woman's beautiful eyes, her beautiful smile, and the velvety long black hair.

Definition 9 – Extended hierarchical image segmentation. Let $\Omega_{roi} \subseteq \Omega$ be an image area part. *Extended hierarchical image segmentation - EHIS -* is a quite new presented approach of image segmentation/classification strategy, which seeks to build a hierarchy of objects in a visual scene. Naturally, we introduce the *EHIS* by **top-down** design-fashion.

First, the *EHIS* hierarchy levels of object topologies will be called *modes*, and they fall into two types:

- **regular mode** - contains the non-essential *objects* and *essential objects*,
- **essential mode** - hierarchical level type, in which only essential objects are.

Further, the *EHIS hierarchical tree* is constructed by recursively *breadth-first searching* objects in a visual scene, which *PCS* segmentator/classifier returns; and node values of *EHIS hierarchical tree* are returned objects $u_1, \ u_2, \ \ldots, \ u_n$ or their *labels* $l_1, \ l_2, \ \ldots, \ l_n$, *EHIS hierarchical tree* will be called *Extended hierarchical image segmentation*, and we will denote it T_{ehis}.

If $\forall l_i \in \mathbb{L}_{\Omega_{roi}} : l_i \in V(T_{ehis})$, where $V(T_{ehis})$ is a vertex set of T_{ehis}, we call T_{ehis} *Complete extended hierarchical image segmentation - CEHIS -*, and denote it T_{cehis}.

In another case, we call T_{ehis} such that

$$V(T_{ehis}) \neq \emptyset$$

Partial extended hierarchical image segmentation - PEHIS -, and denote it T_{pehis}.

Unless otherwise stated, by *Extended hierarchical image segmentation* we mean both *Complete hierarchical image segmentation* and *Partial extended hierarchical image segmentation.*

By introduced theory in the definitions and theorems above, we can clearly define general image segmentation in *semi well-posed* sense.
Let $g_0(x, y) \in G_{im}$ be an image function, which is defined on Ω, $\Omega_{roi} \subseteq \Omega$ be the *Region of Interest*, and $\mathbf{U}_0 = \{u_1, u_2 \dots u_j\}$ be a reasonable *solution-class*. Further, we denote solution, which is returned by a segmentator/classifier, $\widetilde{\mathbf{U}_0}$ as follows: $\widetilde{\mathbf{U}_0} := \{\widetilde{u_1}, \widetilde{u_2} \dots \widetilde{u_i}\}$. If $\widetilde{\mathbf{U}_0} \subseteq \mathbf{U}_0$ or

$$\left(\forall \widetilde{u_i} \in \widetilde{\mathbf{U}_0}\right) \left(\exists! u_j \in \mathbf{U}_0 : \mathbf{C}\left(\widetilde{u_i}\right) \overset{\text{PCS}}{\approx} \mathbf{C}\left(u_j\right)\right), \tag{2.27}$$

we define $\widetilde{\mathbf{U}_0}$ as the **right and *semi-unique* solution** of an appropriate image segmentation/-classification problem. In general, an image segmentation process can be represented as the diagram in **Figure 2.12**.

Figure 2.12: Diagram of image segmentation process.

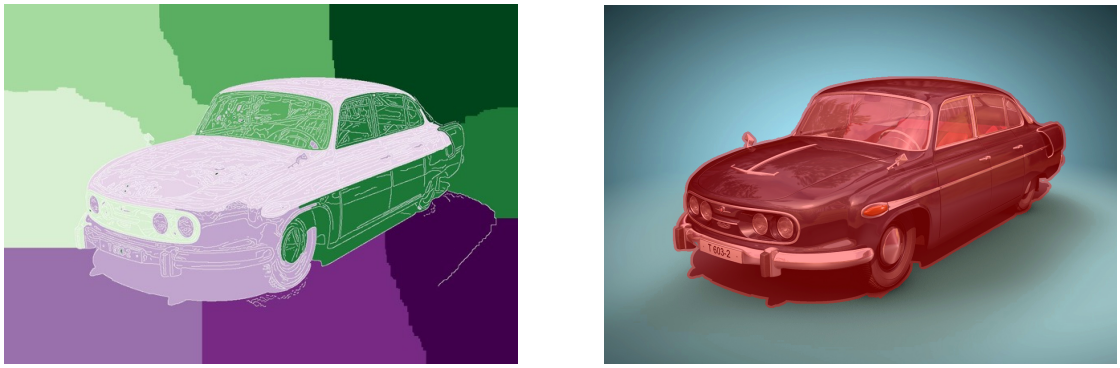
In practical applications, an image segmentator/classifier is typically implemented as a tool-class or a set of functions, which contains operations[4] for image partitioning into suitable regions, merging regions tools, partial solution verificators, etc. Hence, we have to check the well-possedness of used techniques in segmentator/classifier tools too. We mention a few typical categories of image partitioning techniques and their typical representatives in Section 2.3. *À propos*, nowadays, there is no general technique for verifying **Condition 2.27**; therefore, in many cases, the `condition` is visually checked by a human verificator[5].

Further, in the following text, we strictly distinguish ***image partitioning*** and ***image segmentation***, or better ***regular image segmentation***, as follows:

- **image partitioning** - is a reasonable (arbitrary) decomposition of the image scene area Ω into reasonable (arbitrary) regions $\Omega_1, \Omega_2, \dots, \Omega_n$, see **Figure 2.13a**,
- **(regular) image segmentation** - is *PCS* partitioning of the image scene into the object areas $\Omega_1, \Omega_2, \dots, \Omega_i$, and appropriate objects $u_1, u_2, \dots, u_i$ follow **Condition 2.27**, see **Figure 2.13b**.

[4]The operations are based on mathematical approaches for solving typical engineering problems or data-mining techniques.

[5]Only in the simple or straightforward image segmentation/classification tasks, the verification is implemented into an image segmentation/classification computer program.



(a) Image partitioning (b) Image segmentation

Figure 2.13: Image partitioning and segmentation examples.

Inasmuch as presented image segmentation is strictly tied to the problem context, therefore a solution exists in a *solution-class* form (or a subset of a *solution-class*); and as we present in this section, we consider it as the unique solution.
Practical achievement of desired the *PCS* solution is dependent on using an appropriate image partitioning technique or a combination of two or more image partitioning techniques, which has have to be well-posed, respectively. From this point of view, we define the extended image segmentation as the *semi well-posed* problem.

2.2.2 Extended Image Segmentation Theory in examples

From the first reading of the previous subsection, the presented *Extended Image Segmentation Theory* may seem quite complicated, especially, the *solution-class* definition or *object* topology theorem. Hence, in this section, we demonstrate the significant parts of the theory on a few specific examples. We mainly focus on **Theorem 2**, insomuch as it is the direct consequence of **Definition 7**, and *Extended Hierarchial Image Segmentation* definition, which is a direct consequence of **Theorem 2**. Moreover, we cosider **Definition 7**, **Definition 9**, and **Theorem 2** as the integral parts of *Extended Image Segmentation Theory*.

Example 2.1

Let's start with the following simple example, on which we will demostrate *Extended hierarchical image segmentation*, which is described in **Definition 9**.

Let **Figure 2.14** be the input visual scene, an appropriate image function $f(x, y)$ is defined on Ω, $\Omega_{roi} = \Omega$ be the *Region of Interest*, and $\mathbb{L}_{\Omega_{roi}} = \{$`door, estate, garden, (garden) bed, house, lawn, roof, tree` $\}$ be the *label-set*.

Let's set up an image segmentation/classification task: create *PCS* T_{cehis} and an arbitrary T_{pehis}, assign *labels* as the node values and mark up appropriate *modes*.

Figure 2.14: House in the Tudor architectural style.
Picture is referenced from [33].

Solution Let's consider to have the perfect *PCS* segmentator/classifer, as follows: it finds and returns objects in the visual scene by *breadth-first searching* approach, i.e. segmentator/-classifier separately detects object in each *mode* separately. First it finds objects, which are defined on Ω_{roi}, then it marks non-essential objects and it decomposes they in next iteration of image segmentation/classification process. By this approach, it continues until the *mode* is not essential, i.e. in the *mode*, all detected objects are essential.

For our example, the hierarchical *mode* iterations can be as follows:

0^{th} mode iteration In our example, we define the *PCS label-set* as follows: $\Omega_{roi} = \Omega$, and, moreover, $\mathbf{L}\left(f_{|\Omega_{roi}}\right) =$ `estate`. Hence, in this mode iteration, segmentator/classifier returns an object, which is defined in the whole image area Ω, and its appropriate label, i.e. `estate`.
The object $f_{|\Omega_{roi}}$ is obviously not essential, therefore, this *mode* is the regular and segmentator/classifier continuous.

1^{st} mode iteration In the previous iteration, *PCS* segmentator/classifier finds a non-essential object, i.e. `estate`, which, in this iteration, decomposes into a non-essential object, i.e. `garden`, and non-essential object, i.e. `house`. We consider that *PCS* segmentator/classifier is perfect, therefore, the object detection is similar to human perception of the reality. Obviously, the detected objects are not essential, therefore, the segmentator/classifier continous.

2^{nd} mode iteration The segmentator/classifier stepwise extracts from the non-essential *object*, i.e. `garden`, essential objects, i.e. `(garden) bed`, `lawn`, and `tree`, and from the non-essential *object*, i.e. `house`, the essential objects, i.e. `door`, and `roof`, in defined problem context sense. Insomuch as all detected objects are essential, the mode is essential and the segmentator/classifier finishes an image segmentation procedure.

The Appropriate *Complete Hierarchical Extended Image Segmentation* tree T_{cheis} is in the following **Figure 2.15**. For creating an arbitrary *Partial Hierarchical Extended Image Segmentation* tree T_{pheis} is suitable to remove a few nodes and their children, e.g. the node with label `lawn`, and the node with label `door`, see **Figure 2.15**.

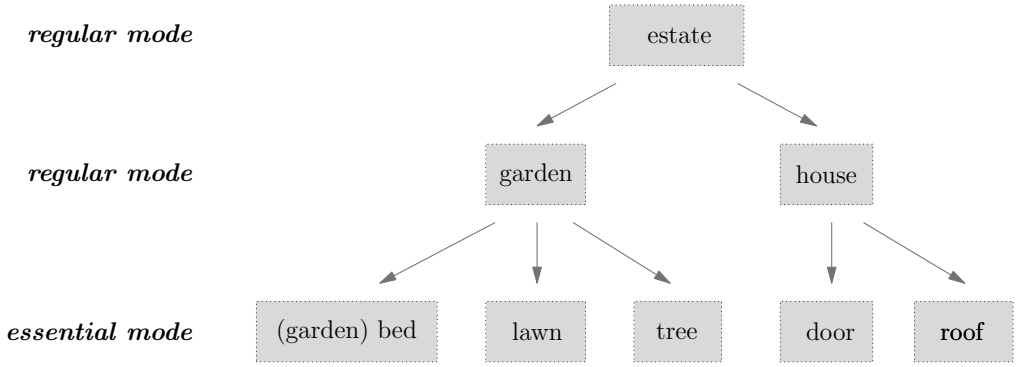


Figure 2.15: T_{cehis} diagram.

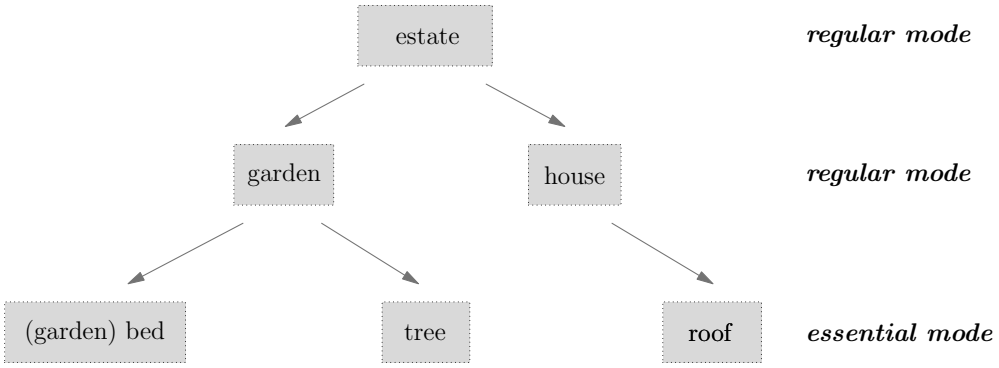
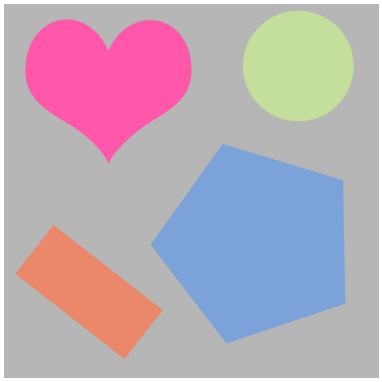


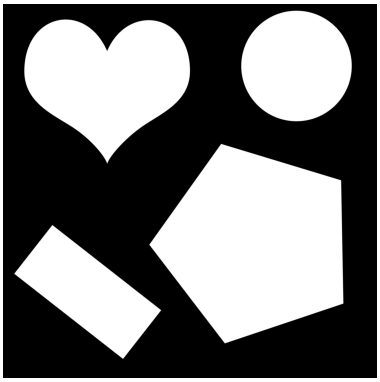
Figure 2.16: T_{pehis} diagram.

Example 2.2

In this and following **Examples 2.3 - 2.4**, we will present object topologies theorem, and *solution-class* definition as the auxiliary definition, on specific image inputs.



(a) Object topologies without contacts.



(b) **Figure 2.17a** binary mask.

Figure 2.17: Example of object topologies without contacts.

Let **Figure 2.17a** be an image input, appropriate image function $f(x, y)$ is defined on image area Ω, **Figure 2.17b** be binary mask of $\Omega_{roi} \subset \Omega$, and $\mathbb{L}_{\Omega_{roi}} = \{\texttt{circle}, \texttt{heart}, \texttt{pentagon}, \texttt{rectangle}\}$ be *label-set*. Let's have the perfect segmentator/classifier then it detects four object $u_1,\ u_2,\ u_3,\ u_4$, which are stepwise defined on $\Omega_1 \subset \Omega_{roi},\ \Omega_2 \subset \Omega_{roi},\ \Omega_3 \subset \Omega_{roi},\ \Omega_4 \subset \Omega_{roi}$, as follows: $\mathbf{L}(u_1) = \texttt{circle}$, $\mathbf{L}(u_2) = \texttt{heart}$, $\mathbf{L}(u_3) = \texttt{pentagon}$, $\mathbf{L}(u_4) = \texttt{rectangle}$. Further, we stepwise denote boundaries of object areas as the $\partial\Omega_1,\ \partial\Omega_2,\ \partial\Omega_3,\ \partial\Omega_4$, and $\mathbb{I} = \{1,\ 2,\ 3,\ 4\}$ as the index-set.
From **Theorem 2** and (2.26), we know that

$$(\forall i, j \in \mathbb{I})(i \neq j) : \Omega_i \subset \Omega_j \ \veebar\ \Omega_i \not\subset \Omega_j.$$

For our example, it is obvious that it applies

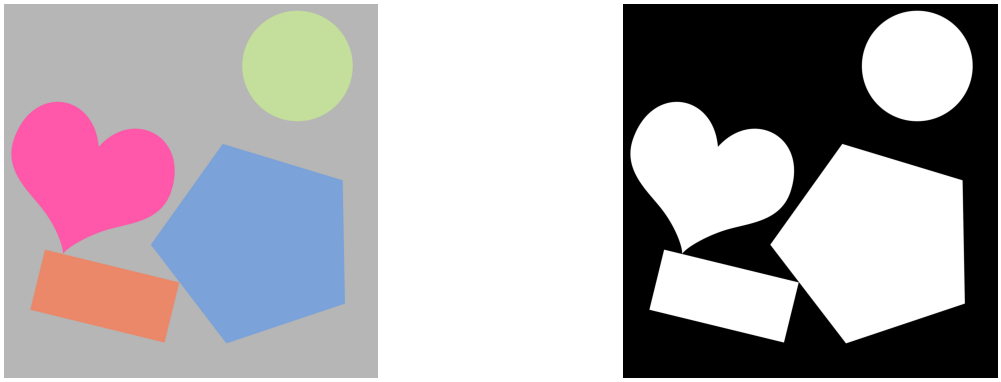
$$(\forall i, j \in \mathbb{I})(i \neq j) : \Omega_i \not\subset \Omega_j,$$

moreover, from **Definition 7** and (2.20), (2.21): let $i, j \in \mathbb{I} : i \neq j$, then

$$\Omega_i \not\subset \Omega_j \Rightarrow \mu(\Omega_i \cap \Omega_j) = 0,$$

$$(\mu(\Omega_i \cap \Omega_j) = 0) \ \wedge\ (\partial\Omega_i \cap \partial\Omega_j = \emptyset) \Rightarrow \mathcal{H}^1(\partial\Omega_i \cap \partial\Omega_j) = 0.$$

Example 2.3



(a) Contacts in two points (b) **Figure 2.18a** binary mask.

Figure 2.18: Example of object topologies with contacts in 2 points.

This example is quite similar to the previous example, but object topologies are slightly changed. Heart and rectangle object areas are in contact in one point, and rectangle and pentagon object areas are in contact in one point too.
Let **Figure 2.18a** be an image input, appropriate image function $f(x, y)$ is defined on image area Ω, **Figure 2.18b** be binary mask of $\Omega_{roi} \subset \Omega$. Let's consider the same *label-set* $\mathbb{L}_{\Omega_{roi}}$ from previous example, and object $u_1,\ u_2,\ u_3,\ u_4$, which are defined in $\Omega_1 \subset \Omega_{roi},\ \Omega_2 \subset \Omega_{roi},\ \Omega_3 \subset \Omega_{roi},\ \Omega_4 \subset \Omega_{roi}$, and $\partial\Omega_1,\ \partial\Omega_2,\ \partial\Omega_3,\ \partial\Omega_4$ are stepwise their boundaries. Let $\mathbb{I}$ be index-set from a previous example.

Inasmuch as the object topologies are slightly changed, it is obvious that it applies the part

$$(\forall i, j \in \mathbb{I})(i \neq j) : \Omega_i \not\subset \Omega_j$$

from **Theorem 2** and (2.26). Moreover, from **Definition 7** and (2.20), (2.21), (2.22):

let $i, j \in \mathbb{I} : i \neq j$, then

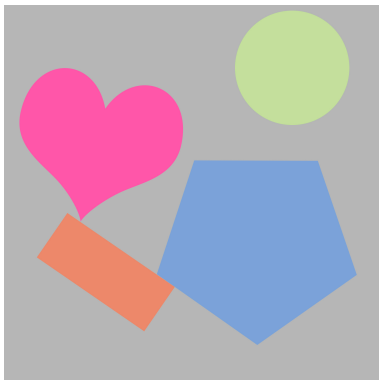
$$\Omega_i \not\subset \Omega_j \Rightarrow \mu\left(\Omega_i \cap \Omega_j\right) = 0,$$

$$\left(\mu\left(\Omega_i \cap \Omega_j\right) = 0\right) \ \wedge \ \left(\partial\Omega_i \cap \partial\Omega_j \neq \emptyset\right) \ \wedge$$

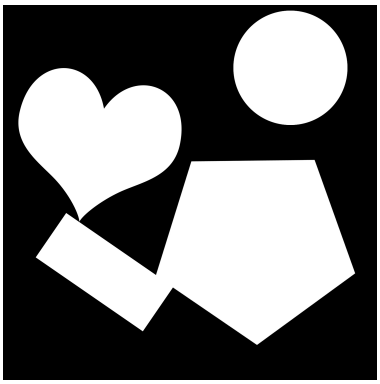
$$\wedge \ \left(\partial\Omega_i \cap \partial\Omega_j \text{ is not Haussdorf 1-measurable}\right) \Rightarrow \mathcal{H}^1\left(\partial\Omega_i \cap \partial\Omega_j\right) = 0.$$

Further, $\mathcal{H}^0\left(\partial\Omega_1 \cap \partial\Omega_2\right) = 0,\ \mathcal{H}^0\left(\partial\Omega_1 \cap \partial\Omega_3\right) = 0,\ \mathcal{H}^0\left(\partial\Omega_1 \cap \partial\Omega_4\right) = 0,$
$\mathcal{H}^0\left(\partial\Omega_2 \cap \partial\Omega_3\right) = 0,\ \mathcal{H}^0\left(\partial\Omega_2 \cap \partial\Omega_4\right) = 1,\ \mathcal{H}^0\left(\partial\Omega_3 \cap \partial\Omega_4\right) = 1$, i.e.
$\sum_{i=1}^{3}\sum_{j=i+1}^{4}\mathcal{H}^0\left(\partial\Omega_i \cap \partial\Omega_j\right) = 2$. It means that the objects are each other in contact in the two points.

Example 2.4



(a) Contacts in one point and boundary part.



(b) **Figure 2.19a** binary mask.

Figure 2.19: Object topologies with contacts in one point and boundary part.

In this case of objects topologies, solution steps are based on examples above, except that rectangle and pentagon relative position is different than their relative position in **Example 2.2**, and, of course, in **Example 2.3**.
As in the previous example, let **Figure 2.19a** be an image input. Let's denote the input image area as the Ω, and an appropriate input image function denote $f(x, y)$; it is defined in Ω. Further, **Figure 2.19b** is the binary mask of *Region of Interest* Ω_{roi}; and $\Omega_{roi} \subset \Omega$. Let $\mathbb{L}_{\Omega_{roi}}$ be the same *label-set* as in the **Examples 2.2 and 2.3**, $u_1,\ u_2,\ u_3,\ u_4$ be objects with the appropriate *labels*, which are same as in the similar examples above. Let $\Omega_1 \subset \Omega_{roi}$, $\Omega_2 \subset \Omega_{roi}$, $\Omega_3 \subset \Omega_{roi}$, $\Omega_4 \subset \Omega_{roi}$ be stepwise their appropriate areas, and the area boundaries are stepwise: $\partial\Omega_1$, $\partial\Omega_2$, $\partial\Omega_3$, and $\partial\Omega_4$. $\mathbb{I}$ is *index-set*, which is same as in the **Example 2.2** and in the **Example 2.3**. Let's denote a common boundary of `pentagon`, i.e. u_3, and `rectangle`, i.e. u_4, object areas as the Γ as follows:

$$\Gamma := \partial\Omega_3 \ \cap \ \partial\Omega_3,$$

and let p_{cont} be a contact point of `heart`, i.e. u_2, and `rectangle` object, i.e. u_4, areas.

In this case, the measure of common object areas are significantly changed. The `pentagon` and `rectangle` object areas have common boundary part Γ. Therefore, **Theorem 2**, **Definition 7**, (2.26), and (2.20), (2.21) imply

$$\Omega_i \not\subset \Omega_j \Rightarrow \mu\left(\Omega_i \cap \Omega_j\right) = 0,$$

$$\left(\mu\left(\Omega_i \cap \Omega_j\right) = 0\right) \ \wedge \ \left(\partial\Omega_i \cap \partial\Omega_j \neq \emptyset\right) \ \wedge$$

$$\wedge \ \left(\partial\Omega_i \cap \partial\Omega_j \text{ is Haussdorf 1-measurable}\right) \Rightarrow \mathcal{H}^1\left(\partial\Omega_i \cap \partial\Omega_j\right) = \Gamma.$$

Further, $\mathcal{H}^0(\partial\Omega_1 \cap \partial\Omega_2) = 0,\ \mathcal{H}^0(\partial\Omega_1 \cap \partial\Omega_3) = 0,\ \mathcal{H}^0(\partial\Omega_1 \cap \partial\Omega_4) = 0,$
$\mathcal{H}^0(\partial\Omega_2 \cap \partial\Omega_3) = 0,\ \mathcal{H}^0(\partial\Omega_2 \cap \partial\Omega_4) = 1,\ \mathcal{H}^0(\partial\Omega_3 \cap \partial\Omega_4) = +\infty$, i.e.
$\sum_{i=1}^{3}\sum_{j=i+1}^{4}\mathcal{H}^0(\partial\Omega_i \cap \partial\Omega_j) = +\infty$. It means that the objects are each other in contact in Haussdorf 1-measurable part, i.e. Γ, which is mentioned above.

2.3 Image Partitiong Methods

In general, we study only theoretical approaches on image segmentation and partition techniques for the time being. But, there are methods for practical purposes and their implementations. We can divide these methods into a several groups as follows: *intensity based methods, discontinuity based methods, similarity based methods, clustering methods, graph based methods, pixon based methods, hybrid methods.* For other informations about the methods, see e.g. [34].

However, neither techniques is general to proceed arbitrary image functions. It implies from **Kleinsberg's Imposibility theorem** as follows: *there is not a general function for the general image segmentation and recognition problems* [35].

In the next two chapters, we study two, commonly known, image clustering techniques, i.e. *Lloyd type image clustering method* and *image spectral clustering method.*

Chapter 3

Lloyd Type Image Clustering Methods

In general, clustering techniques, which are derivates of Lloyd clustering algorithm, belong to relatively simple image partitioning methods. These techniques do not treat any specific properties for image functions, e.g. functions have to be L^2; they only work with a range of values of input image discrete functions. The significant problem of these clustering approaches is partitioning an input image, see **Figure 3.1a**, into regions, which are not compact, see **Figure 3.1b**. Hence, these techniques are often used as the auxiliary procedures in more sophisticated image segmentation methods.

(a) Input image. (b) Appropriate clustered image.

Figure 3.1: Example of using clustering image partitiong technique the k-means++. The picture is referenced from [36].

3.1 Lloyd type clustering techniques

Lloyd Type Clustering Techniques are based on the idea of assigning data into so-called Voronoi cells, which are commonly denoted as the clusters, so that they minimize a sum of squares, squared Euclidean distance is typically used, within cell. The standard algorithm

was first introduced by Stuart P. Lloyd, Bell Laboratories employee, in 1957 as a pulse-code modulation technique [37], but it was not published outside Bell Laboratories until 1982 [38]. Hence, in computer science and electrical engineering, the standard algorithm is commonly known as Lloyd's algorithm or Voronoi iteration.

In 1967, James MacQueen first used the term `k-means` in his work *Some Methods for classification and Analysis of Multivariate Observations* [39], which continues to Steinhaus's work *Sur la division des corps matériels en parties* [40].
In computer science field, especially in image processing community, MacQueen's k-means is related to Lloyd algorithm, however they differ in appropriate inputs: Lloyd algorithm input is a continuous geometric region whereas MacQueen's k-means require discrete points in $\mathbb{R}^n$, where $n \in \mathbb{N}$, as the input data set. Hence, our inputs are discrete points belonging to $\mathbb{R}$, i.e. in case of grayscale images, or they belong to $\mathbb{R}^3$, i.e. in case of colour images. Hence, we focus on k-means algorithm and algorithms, which are derived from the k-means.

3.1.1 Algorithm k-means

Partitioning point set input data into k subsets commonly called clusters, yet relabelled as regions in this text, is *NP*-hard problem. Therefore, the k-means algorithm was proposed as the optimization solver for the *NP*-hard problem mentioned above.

Let's stepwise denote input points, refered to as pixel values in our text, $p_1,\ p_2,\ \ldots,\ p_n$, belonging to input set P, and desired k regions $S_1,\ S_2,\ \ldots,\ S_k$, beloging to region set S. Often, the k-means is denote as *hard*-clustering type algorithm, i.e.

$$(\forall p_i \in P)\,(\exists! S_j \in S : p_i \in S_j)\,.$$

It quite corresponds to our desired properties of image objects, which we denote in the previous chapter, i.e. an arbitrary point p of image function area Ω, and in each mode iteration the point $p \notin \partial\Omega$ corresponds to one object or *semi-essential* object only. As we say at the beginning of this chapter, the Lloyd type clustering techniques generate regions which are not compact, therefore they cannot correspond to any objects. By our definitions in the previous chapter, the k-means algorithm is an image partitioning technique, which decomposes an image area into reasonable non-compact regions in most cases.

As we say, the k-means is an optimizer, hence, the solution, which is returned by this solver, is a relax of the optimal solution. Thus, let $c_1,\ c_2,\ \ldots,\ c_k$ be centeroids of the aprropriate regions $S_1,\ S_2,\ \ldots,\ S_k$. As we note, the regions are build, so that they minimize a sum of squares within each region, i.e. the k-means solves the minimazition problem

$$\underset{S}{\operatorname{argmin}} \sum_{j=1}^{k} \sum_{p \in S_j} \|p - c_j\|_2^2. \tag{3.1}$$

In general, we can divide k-means algorithm procedure into two significant steps, i.e. an **initialization step** and an **update step**, as follows:

initialization step The algorithm determines the nearest centroid from the set of centroids $\{c_1,\ c_2,\ \ldots,\ c_k\}$, which we denote C, of the appropriate regions $S_1,\ S_2,\ \ldots,\ S_k$ for each input point $p \in P$ as follows: *it stepwise compares the Euclidean distance between the point and each centroid $c \in C$, and the point is assigned to region $S_j \in S$, in which the Euclidean distance between the point and appropriate centroid is minimal.* As the starting initialization is typically random assignment of points $p_1,\ p_2,\ \ldots,\ p_n$ to regions $S_1,\ S_2,\ \ldots,\ S_k$ or a random selection of centroids from the input point set P and then an assignment of input points by the approach mentioned above. Mersenne twister [41] is typically used as the *pseudo-random* integer generator.

update step The algorithm computes new coordinates of region centroids as an arithmetic mean of the points within an appropriate region. Since the arithmetic mean is a *least-squares* estimator, it minimizes the within-region sum of squares.

These two steps are repeated until the maximum iterations is exceeded or region changes, i.e. point region membership changes, are relatively small. By following pseudocode, we simply describe the entire the k-means procedure.

Algorithm 1: k-means

Input : input set $P = \{p_1,\ p_2,\ \ldots,\ p_n\}$, desired region count k, reasonable small $\varepsilon \in \mathbb{R}$, maximum iteration count M
Output: regions $S_1,\ S_2,\ \ldots,\ S_k$

1. Randomly choose k starting centroids from P and assign them to starting centroid set C^0.
2. Assign points to initial regions $S_1^0, S_2^0, \ldots S_k^0$ by their *Euclidean distance* to te nearest centroid $c_j^0 \in C^0$ and set the initial memership vector U^0 as follows:
   for $p_i \in P$ $U(i) = j^0$, where i is index of the point and j^0 is the nearest centroid index.
3. Set iteration counter $b = 0$ and $\delta = 0$.
4. Compute new region centroids as the arithmetic mean of points within-region, i.e.
$$c_j^{t+1} = \frac{1}{|S_j^t|} \sum_{p_i \in S_j^t} p_i$$
5. For each point $p_i \in P$ find the nearest centroid $c_j^{t+1} \in C^{t+1}$ by the *Euclidean distance* between them. Further, assign points to appropriate regions $S_1^{t+1},\ S_2^{t+1},\ \ldots,\ S_k^{t+1}$ and set membership vector U^{t+1}; if $U(i)^{t+1} = U(i)^t$ do nothing, otherwise $\delta = \delta + 1$.
6. If $\frac{\delta}{|P|} < \varepsilon$ or $b > m$ finish, otherwise set $\delta = 0$ and $b = b + 1$ and continue to step 4.

In practical implementation of the k-means algorithm, a point assignment to an appropriate region is implemented as follows: the point memberships are only stored in membership vector U. This approach is efficient for a computer memory management, because there is no point region reassignment overload.

3.1.2 Algorithm k-means++

The k-means algorithm quickly converges to a local optimum of potential function ϕ, but there are many practical examples for which it generates arbitrarily bad clustered regions, i.e. $\frac{\phi}{\phi_{opt}}$ is unbounded even when $n := |P|$ and k are fixed. It shows that the causes of this fact are the random choice of starting centroids and appropriate assignment input points to starting regions.

Therefore, in 2007, David Arthur and Sergei Vassilvitskii proposed an efficient technique for selecting k-means starting centroids from an input point set P, and they called this technique k-means++. Its general principle is uniformly random choice of starting centroids from input point set at probability distribution, which is computed as the squared *Euclidean distance* to the centroid that has already been chosen, as follows:

Algorithm 2: k-means++

Input : input set $P = \{p_1,\ p_2,\ \ldots,\ p_n\}$, desired region count k, reasonable small $\varepsilon \in \mathbb{R}$, maximum iteration count M
Output: regions $S_1,\ S_2,\ \ldots,\ S_k$

1. Uniformly randomly choose a centroid c_1^0 from the input point set and assign c_1^0 to the set of centroids C.
2. For each $p \in P$, compute $\mathrm{D}(x)$ as the *Euclidean distance* between p and the centroid that has been already chosen.
3. Choose next starting centroid c_j^0 randomly at probability distribution $\mathrm{D}(x)^2$.
4. Repeat 1 and 2 until the k starting centroids have been chosen, then continue with the standard k-means algorithm.

By this augmentation of the k-means, the the algorithm k-means++ becomes $\mathcal{O}(\log k)$-competitive, where k is the desired region count. Moreover, it can be proved that $E[\phi] \leq 8(\ln k + 2)\,\phi_{opt}$, i.e. the solution, which is returned by a solver, is at most $8(\ln k + 2)$ worse by factor then optimal solution.

3.1.3 Regular k-means+2 design-fashion

In practical application, e.g. image partitioning applications, which use spectral clustering method, it shows that the k-means++ upper bound, i.e. $8\,(\ln k + 2)\,\phi_{opt}$, is still too high. Hence, we introduce a new design-fashion approach for choosing starting centroids $c_1^0,\ c_2^0,\ \ldots,\ c_k^0$. We called these techniques *regular k-means*+2.

If we desire high accuracy clustered regions, it shows that it is better to choose starting centroids $c_1^0,\ c_2^0,\ \ldots,\ c_k^0$, which are most distant from each other, instead of choosing these centroids at randomly at distribution, which we note in the previous section. By this initialization approach, the k-means iteration can finish in 2 iterations in a specific application set. If we do not take into account the first initial iteration, the k-means iteration can finish in 1 standard iteration, indeed. To obtain centroids $c_1^0,\ c_2^0,\ \ldots,\ c_k^0$, which fulfilling the properties above, we propose the following approach:

Algorithm 3: regular k-means+2 design-fashion

Input : input set $P = \{p_1,\ p_2,\ \ldots,\ p_n\}$, desired region count k, reasonable small $\varepsilon \in \mathbb{R}$, maximum iteration count M
Output: regions $S_1,\ S_2,\ \ldots,\ S_k$

1. Uniformly randomly choose a centroid c_1^0 from the input point set and assign c_1^0 to the set of centroids C,
2. For each $p_i \in P$, compute the *Euclidean distances* $\rho_1,\ \rho_2,\ \ldots,\ \rho_n$ among the each chosen centroids $c_1^0,\ c_2^0,\ \ldots,\ c_j^0 \in C$, and determine $p_i \in P$ the *Euclidean distance*, i.e. ρ^i, to the each chosen centroids $c_1^0,\ c_2^0,\ \ldots c_j^0,\ \in C$ as the minimal value from $\rho_1,\ \rho_2,\ \ldots,\ \rho_j$.
3. From $\rho^1,\ \rho^2,\ \ldots,\ \rho^n$, where $n := |P|$, choose maximum value and the appropriate point $p \in P$ assign to C.
4. Repeat 2 and 3 until the k centroids have been choosen, then continue with the standard k-means algorithm.

In general, the steps 1 and 2 solve *Minimax* problem, i.e. the most distant, i.e. point from fixed points, i.e. choosen centroids in our case, is the point, which has maximum distance value from $\rho_1,\ \rho_2,\ \ldots,\ \rho_n$, i.e. minimal distances of points $p_1,\ p_2,\ \ldots,\ p_n$ to fixed points.

In this subsection, we introduce design-fashion of new initialization approach for k-means clustering method. We would like to note that proper analysis of the algorithm is beyond the scope of this work and we are going to focus on this analysis in the future.

3.2 Example

At the end of this chapter, we prepare an example of using our method **regular k-means+2** in a so-called colour compression technique. The colour compression is based on a reduction of colour information, i.e. color count, in an input picture; it implies that file size is decreased. We test the **regular k-means+2** on large picture, its size is 4188x2792px, see **Figure 3.2a**.

We set up $k = 500$ as the desired regions count. The achieved result is shown in **Figure 3.2b**. Moreover, the image size is decreased from 3.3 MB to 2.3 MB, what we consider as a good result.

(a) Input picture. (b) Colour compressed picture.

Figure 3.2: Colour compression example.
Picture is referenced from [42].

Chapter 4

Image Spectral Clustering Methods

In the last three decades, spectral clustering becomes widely used technique for exploratory data analysis with applications ranging from statistic [43], electrical engineering [44], biology [45], psychology [46], etc. Therefore, it is not surprise that image processing communities conduct extensive research, which is focused on application of spectral clustering as the image partitioning technique, in the last two decades, see e.g. [47], [48], [49]. In our opinions, there are many unanswered questions, e.g. the general representation of the *Laplace matrix* spectrum and its relationship to region geometries and topologies in an image scene. Moreover, there is no general, commonly known approach to build spectral embedding for input data, i.e. pixels in our case. A few partial answers to these questions can be found in [50], therefore, we draw on the presented knowledge and try to generalize it.
We realize that the issues are quite complicated and a few of them require interdisciplinary mathematical approaches to understand their significant mathematical properties. Therefore, it may seem that our explanations are not related to image partitioning techniques, but it is not true. Interdisciplinary explanations are typical approaches in image processing communities.
In specific image processing applications, images and the related procedures can be described from different theory point of views, e.g. the graph theory, mechanical engineering theory, signal processing theory and others.

(a) Santa María del Fiore.

(b) Dr. David Horák.

Figure 4.1: Image spectral clustering examples.

Therefore, the principles of spectral clustering theory go back to Donath and Hoffman [51], in which they proposed the partition graphs based on eigenvectors of their adjacency matrix in 1973. Another significant discover, can be found in [52], which was published by Fiedler in the same year.

In this work, Fiedler focused on the properties of the *graph Laplacian* eigenvector corresponding to the second smallest eigenvalue and its connection to bi-graph partitioning. This eigenvector closely corresponds to the second lowest string mode, and it is often called *Fiedler eigenvector* of *graph Laplacian*. Since then, the spectral clustering theory was discovered, rediscovered, and extended many times in different scientific communities, e.g. [53], [54], [55].
In 2007, Luxburg published an article called *A tutorial on spectral clustering* [56], in which the essential spectral clustering theory is summarized, and, nowadays, it belongs to the fundamental works of modern approaches to this theory.

In this chapter, we continue with one of the image spectral clustering theory, which is introduced in [47]. However, we mainly focus on the lowest part of Laplacian matrix spectrum and its relationships to region geometries and topologies in an image scene. The integral parts of this theory, i.e. relations to the graph theory, the basic *Laplacian matrix* properties, and problem formulations, are referenced from [56].

4.1 Fundamental mathematical model and notations

In [56, p. 1], there is introduced a mathematical modelling approach, which is based on the graph theory. Relations among input data points are modelled by the similarity graph, where the data points are represented as the graph vertices and appropriate relation strengths among them as the edge weights. Afterwards, the input data set decomposition into reasonable subsets, i.e. input image pixel assignment to reasonable regions in our case, is formulated as the graph minimum-cut/max-flow problem [57, pp. 120–128]: *we desire to find such graph partitions with edges within the partitions having high weight, and simultaneously with edge between two different graph partitions is very low.*.

4.1.1 Graph notation

Let $G = (V, E)$ be an undirected graph with a vertex set $V = \{v_1,\ v_2,\ \ldots,\ v_n\}$ and an edge set $E = \{e_1,\ e_2,\ \ldots,\ e_m\}$. Let's assume that the graph G is weighed and $\mathbf{W} \in \mathbb{R}^{n\times n}$, where $n := |V|$, is its *adjacency matrix*, in which all cells have non-negative values, i.e. $\forall i, j \in \{1,\ 2,\ \ldots,\ n\} : w_{ij} \geq 0$. A degree d_i of a vertex $v_i \in V$ is defined as follows

$$d_i = \sum_{j=1}^{n} w_{ij}.$$

Let $\mathbf{D} \in \mathbb{R}^{n \times n}$ be a graph degree matrix with the values $d_1,\ d_2,\ \ldots,\ d_n$ on the main diagonal. Furter, let $A \subseteq V$. For A, we define a characteristic vector $\mathbf{e}_A \in \mathbb{Z}^n$ such that

$$v_i \in V, \quad \mathbf{e}_A(v_i) := \begin{cases} 1 & \text{if a vertex } v_i \in A, \\ 0 & \text{otherwise.} \end{cases}$$

Let $G_1,\ G_2,\ \ldots,\ G_k \subset G$ be connected components. We stepwise denote vertex sets $V(G_1)$, $V(G_2), \ldots, V(G_k) \subset V$ as the $A_1,\ A_2,\ \ldots,\ A_k$. The sets $A_1,\ A_2,\ \ldots,\ A_k$ form a graph partition if

$$\forall i, j \in \{1, 2, \ldots, k\}, i \neq j : V(A_i) \cap V(A_j) = \emptyset, \tag{4.1}$$

$$\bigcup_{i=1}^{k} V(A_i) = V. \tag{4.2}$$

For the spectral clustering theory explanations, it is suitable to define following two measurements of set, i.e.

$$|A| := \text{ vertex count in A} \tag{4.3}$$

and

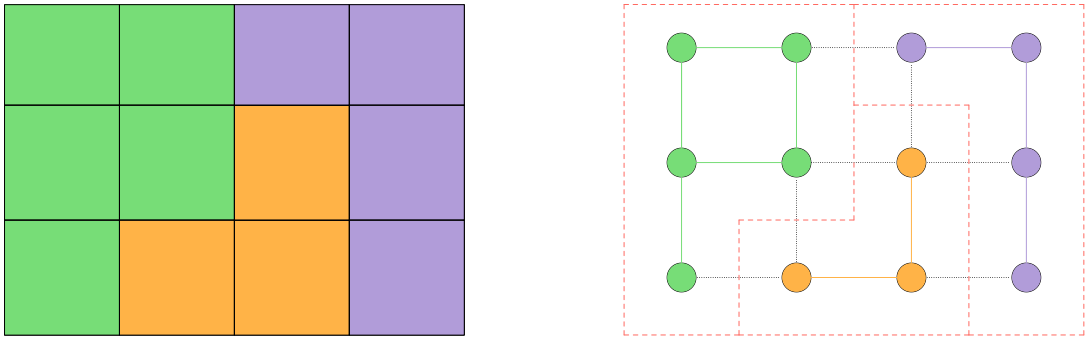
$$\operatorname{vol}(A) := \sum_{i=1}^{n} \mathbf{e}_A(v_i) \cdot d_i. \tag{4.4}$$

In practical applications, instead of *continuous image functions* the *discrete image functions* are typically processed. The continuous image approaches have only academic significance for introducing the image processing theories, as we know.
Therefore, let's notice that presented graph partitioning partially corresponds to a few of *Extended image segmentation theory* claims as their discrete approximations. For example, we require the object area have to be compact, and the presented graph partitions are connected components, i.e. from each point belonging to an appropriate connected component, there are paths to other points in the same component. Therefore, we can consider connected components as closed sets in a discrete sense, and, moreover, they are bounded by default.[a] Hence, we can consider connected components as compacts in this point of view. See **Figure 4.2**.
On the other hand, the (4.3) can be regarded as a discrete analogue of *Lebesgue measure*, which we use for determination of object area sizes in Chapter 2.
In the next section, we present more information about constructing image similarity graphs.

[a] It is obvious, because required input image functions are discrete.

4.1.2 Constructing image similarity graphs

In [56, p. 2], 3 types of similarity graphs, i.e. ϵ-neighborhood, k-nearest neighbor, and fully connected, are presented. Unfortunately, neither of them is applicable for our purposes by their presented definitions.



(a) Input image matrix. (b) Appropriate minimal graph cut.

Figure 4.2: Image partitioning as a graph decomposition problem.

It is suitable to require an image-graph to have its sturcture corresponding to natural digital image discretization, in which a discretization step is equaled 1. For these requirements, a *grid graph* [58], which is commonly known as a *lattice* or *mesh*, is a good candidate. By row, we stepwise denote image pixels as $p_1,\ p_2,\ \ldots,\ p_n$, where n is equaled an image resolution, see **Figure 4.3**.

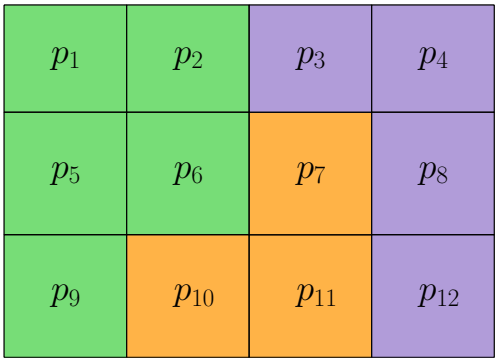


Figure 4.3: Image pixels numbering.

Then, we map pixel set $P := \{p_1,\ p_2,\ \ldots,\ p_n\}$, to graph vertex set $V := \{v_1,\ v_2,\ \ldots,\ v_n\}$ as follows:

$$p_1 \mapsto v_1,\ p_2 \mapsto v_2,\ \ldots,\ p_n \mapsto v_n. \tag{4.5}$$

Further, in image processing field, the *Gaussian function* is typically used for determining similarity between two pixels, i.e. p_i and p_j, v_i and v_j respectively, which are connected by edge in a similarity graph G. The *Gaussian function*, sometimes called the *Gaussian kernel similarity function* or *Gaussian*, is defined such that

$$\mathcal{S}_{gauss}\left(p_i, p_j\right) := \exp\left(\frac{\|p_i - p_j\|_2^2}{2\sigma^2}\right), \tag{4.6}$$

where p_i, $p_j \in P$ and $\sigma \in \mathbb{R}$ is a *variance* parameter, which we call a *detailness* parameter. The typical *Gaussian kernel function* graph is shown in **Figure 4.4**.

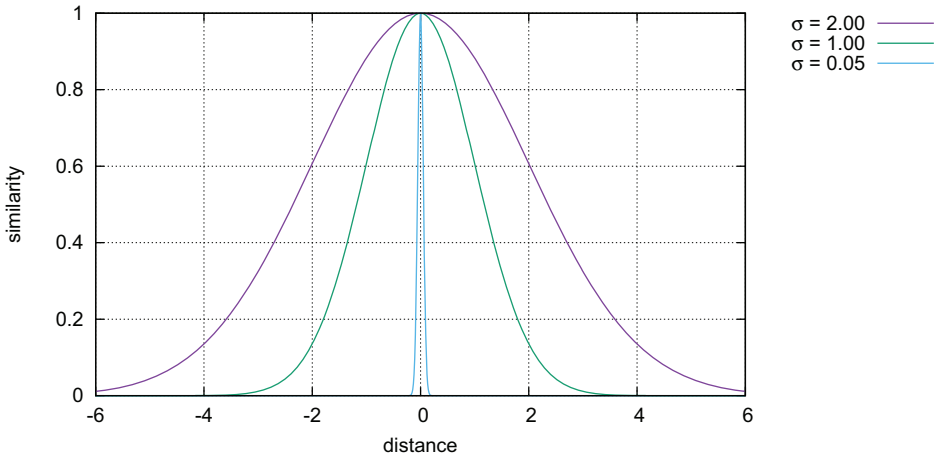


Figure 4.4: Gaussian kernel function.

4.2 Graph Laplacians and their basic properties

In this section, we define several types of graph Laplacian matrices and their fundamental properties, which play significant roles in spectral clustering theory. We point out main ideas, which are introduced in [56, pp. 3 - 5].
Moreover, there is a whole field, which is dedicated to study these matrices, and it is called the *spectral graph theory*. For further information, see e.g. [59].

Further, for an arbitrary characteristic vector $\mathbf{e}_A \in \mathbb{Z}^n$, where $n = |V|$ and $A \subseteq V$, it is suitable to define an equivalence class Ξ_A such that

$$\forall a \in \mathbb{C} \setminus \{0\} : a \cdot \mathbf{e}_A \in \Xi_A, \tag{4.7}$$

moreover, it applies

$$\forall \widetilde{\mathbf{e}_A} \in \Xi_A : \mathbf{L}\widetilde{\mathbf{e}_A} = \mathbf{o}, \tag{4.8}$$

where $\mathbf{L}$ is an *unnormalized graph Laplacian*. Last proposition, i.e. (4.8), is direct consequence of the properties of *unnormalized graph Laplacian* matrix. The characteristic vector $\mathbf{e}_A$ of subset A is typically chosen as the representative of vectors $\widetilde{\mathbf{e}_A} \in \Xi_A$.

Unnormalized graph Laplacian

Definition 10 – Unnormalized graph Laplacian L. Let $G = (V, E)$ be weighed undirected graph, $\mathbf{W}$ be its *adjacency matrix* as follows $\forall i, j \in \{1,\ 2,\ \ldots,\ |V|\} : w_{ij} \geq 0$, and $\mathbf{D}$ be its *degree matrix*. Then, the *unnormalized graph Laplacian* is defined such that

$$\mathbf{L} := \mathbf{D} - \mathbf{W}. \tag{4.9}$$

Proposition 1 – Properties of L. The matrix $\mathbf{L}$ satisfies the following properties:

1. For every vector $\mathbf{f} \in \mathbb{R}^n$, it applies
$$\mathbf{f}^{\mathrm{T}}\mathbf{L}\mathbf{f} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\left(f_i - f_j\right)^2. \tag{4.10}$$
2. $\mathbf{L}$ is symmetric and positive semi-definite, which we denote *SPS*.
3. The smallest eigenvalue of $\mathbf{L}$ is 0. If multiplicity of zero eigenvalue is equaled 1, then appropriate eigenvector is $\mathbf{e} = (1,\ 1,\ \ldots,\ 1)$.
4. $\mathbf{L}$ has n non-negative, real-valued eigenvalues, i.e. $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.

The proof of **Proposition 1** can be found in [56, p. 3]. Other interesting properties of the *unnormed Laplacian matrix* are presented in [60].

Proposition 2 – Number of connected components of graph G and matrix $\mathbf{L}$ lowest magnitude spectrum. Let $G = (V, E)$ be an undirected graph with non-negative weights. Then multiplicity k of the eigenvalue 0 of $\mathbf{L}$ equals the number of connected components $A_1 \subset V,\ A_2 \subset V,\ \ldots,\ A_k \subset V$ in the graph G. The appropriate eigenvectors for 0 eigenvalues are characteristic vectors $\mathbf{e}_{A_1},\ \mathbf{e}_{A_2},\ \ldots,\ \mathbf{e}_{A_k}$ of these components.

The **Proposition 2** proof is introduced in [56, p. 4].

Normalized graph Laplacians

In literature, e.g. [59] or our standard reference [56], two normalized Laplacian matrices are introduced.

Definition 11 – Normalized graph Laplacian $\mathbf{L_{sym}}$ and $\mathbf{L_{rw}}$. Let $G = (V, E)$ be weighed undirected graph, $\mathbf{W}$ be its *adjacency matrix* as follows $\forall i, j \in \{1,\ 2,\ \ldots,\ |V|\} : w_{ij} \geq 0$, and

$\mathbf{D}$ be its *degree matrix*. Further, we define two normalized graph Laplacian matrices such as

$$\mathbf{L}_{sym} := \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}} \tag{4.11}$$

and

$$\mathbf{L}_{rw} := \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}, \tag{4.12}$$

where $\mathbf{I}$ is an *identity matrix*.

The matrices $\mathbf{L}_{sym}$ and $\mathbf{L}_{rw}$ are closely related to each other. $\mathbf{L}_{rw}$ is often interpreted as the *Markov matrix* and its eigenvectors as the *random paths* [61]. On the other hand, $\mathbf{L}_{sym}$ is typically interpreted as $\mathbf{L}_{rw}$ symmetric version.

Proposition 3 – Properties of $\mathbf{L}_{sym}$ and $\mathbf{L}_{rw}$. The *normalized Laplacians* satify the following properties:

1. For every vector $\mathbf{f} \in \mathbb{R}^n$, it applies

$$\mathbf{f}^{\mathrm{T}}\mathbf{L}\mathbf{f} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}}\right)^2. \tag{4.13}$$

2. λ is an eigenvalue of $\mathbf{L}_{rw}$ with an appropriate eigenvector $\mathbf{u}_{rw}$ if and only if λ is an eigenvalue of $\mathbf{L}_{sym}$ with an appropriate eigenvector $\mathbf{u}_{sym} = \mathbf{D}^{\frac{1}{2}}\mathbf{u}_{rw}$.

3. λ is an eigenvalue of $\mathbf{L}_{rw}$ with an appropriate eigenvector $\mathbf{u}$ if and only if λ and $\mathbf{u}$ solve the generalized eigenproblem $\mathbf{L}\mathbf{u} = \lambda\mathbf{D}\mathbf{u}$.

4. If multiplicity of an zero eigenvalue of L_{rw} is equaled 1, the approriate eigenvector is $\mathbf{e} = (1, 1, \ldots 1)$. If 0 is eigenvalue of $\mathbf{L}_{sym}$ with multiplicity 1, the approriate eigenvector is $\mathbf{D}^{\frac{1}{2}}\mathbf{e}$.

5. Both $\mathbf{L}_{sym}$ and $\mathbf{L}_{rw}$ are positive semi-definite and have n non-negative real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.

The appropriate proof of **Proposition 3** is introduced in [56, p. 5].

Proposition 4 – Number of connected components of graph G, and matrices $\mathbf{L}_{rw}$ and $\mathbf{L}_{sym}$ lowest magnitude spectra. Let $G = (V, E)$ be an undirected graph with non-negative weights. Then the multiplicity k of the eigenvalue 0 of both $\mathbf{L}_{rw}$ and $\mathbf{L}_{sym}$ equals the number of connected components $A_1, A_2, \ldots, A_k$ in the graph G. For $\mathbf{L}_{rw}$, the appropriate eigenvectors for zero eigenvalues are characteristic vectors $\mathbf{e}_{A_1}, \mathbf{e}_{A_2}, \ldots, \mathbf{e}_{A_k}$ these components. For $\mathbf{L}_{sym}$, the eigenspace of 0 is spanned by the eigenvectors $\mathbf{D}^{\frac{1}{2}}\mathbf{e}_{A_1}, \mathbf{D}^{\frac{1}{2}}\mathbf{e}_{A_2}, \ldots, \mathbf{D}^{\frac{1}{2}}\mathbf{e}_{A_k}$.

For proof of the proposition above, we refer to standard reference [56, p. 5].

4.3 Problem formulations

Based on previous definitions, propositions, and notes, we present three fundamental algorithms, i.e. unnormalized and two normalized spectral clustering algorithms, which are introduced in our standard reference [56, pp. 6-7]. For our purposes, we slightly redesign them inasmuch as these standard algorithms are quite general, and do not correspond well to image partitioning problems. Then, we re-formulate image spectral algorithms into graph cut point of views.

4.3.1 Standard image spectral clustering algorithms

Let's begin with the basic *unnormalized image spectral clustering* technique.

Algorithm 4: Unnormalized image spectral clustering

Input : digital image Im, detailness parameter $\sigma \in R$, desired region count k, reasonable small $\varepsilon \in \mathbb{R}^+$, maximum iteration count M
Output: regions $S_1, S_2, \ldots, S_k$

1. By row, stepwise denote Im pixels as the $p_1,\ p_2,\ \ldots,\ p_n$, where n is equaled Im resolution, and map them to a vertex set V, see (4.5).
2. Construct an image similarity grid-graph $G = (V, E)$, and use σ as a detailness parameter for the *Gaussian kernel similarity function*. Let $\mathbf{W}$ be its *adjacency matrix*, $\mathbf{D}$ be its *degree matrix*.
3. Compute *unormalized graph Laplacian* $\mathbf{L} = \mathbf{W} - \mathbf{D}$.
4. Compute the first k eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k$ of *unormalized graph Laplacian* $\mathbf{L}$.
5. Let $\mathbf{X} \in \mathbb{R}^{n \times k}$ be matrix, which contains the vectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k$ as columns.
6. Remap Im pixels $p_1,\ p_2,\ \ldots,\ p_n$ to $\mathbb{R}^k$ such that

   $$p_1 \mapsto \mathbf{x_1},\ p_2 \mapsto \mathbf{x_2},\ \ldots,\ p_n \mapsto \mathbf{x_n}$$

   where x_i is i-th rows of matrix $\mathbf{X}$.
7. By one of the *Lloyd type clustering algorithms*, cluster points $\mathbf{x_1},\ \mathbf{x_2},\ \ldots,\ \mathbf{x}_n$ to regions $C_1,\ C_2,\ \ldots,\ C_k$. The Lloyd type algorithm maximum of iterations is equaled M, and ε is stop criterium for the k-means iterations.
8. Map $C_1,\ C_2,\ \ldots,\ C_k$ to $S_1, S_2, \ldots, S_k$ such that

   $$(\forall j \in \{1, 2, \ldots, k\})(\forall \mathbf{x}_i \in C_j)(\mathbf{x}_i \mapsto p_i \text{ and } p_i \text{ assign to } S_j)$$

The following two *normalized* algorithms are redesigned approaches, which their standard variants are firstly presented in *Shi and Malik* article [47], *Ng, Jordan, Weiss* article [62].

Algorithm 5: Normalized image spectral clustering (Shi and Malik)

Input : digital image Im, detailness parameter $\sigma \in R^+$, desired region count k, reasonable small $\varepsilon \in \mathbb{R}$, maximum iteration count M
Output: regions $S_1, S_2, \ldots, S_k$

1. By row, stepwise denote Im pixels as the $p_1,\ p_2,\ \ldots,\ p_n$, where n is equaled Im resolution, and map them to a vertex set V, see (4.5).
2. Construct an image similarity grid-graph $G = (V, E)$, and use σ as a detailness parameter for the *Gaussian kernel similarity function*. Let $\mathbf{W}$ be its *adjacency matrix*, $\mathbf{D}$ be its *degree matrix*.
3. Determine *unormalized graph Laplacian* $\mathbf{L} = \mathbf{W} - \mathbf{D}$.
4. Compute the first k eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k$ of the generalized eigenproblem, i.e
$$\mathbf{Le} = \lambda \mathbf{De}.$$
5. Let $\mathbf{X} \in \mathbb{R}^{n \times k}$ be matrix, which contains the vectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k$ as columns.
6. Remap Im pixels $p_1,\ p_2,\ \ldots,\ p_n$ to $\mathbb{R}^k$ such that
$$p_1 \mapsto \mathbf{x}_1,\ p_2 \mapsto \mathbf{x}_2,\ \ldots,\ p_n \mapsto \mathbf{x}_n$$
where $\mathbf{x}_i$ is i-th row of matrix $\mathbf{X}$.
7. By one of the *Lloyd type clustering algorithms*, cluster points $\mathbf{x}_1,\ \mathbf{x}_2,\ \ldots,\ \mathbf{x}_n$ to regions $C_1,\ C_2,\ \ldots,\ C_k$. Lloyd type algorithm maximum of iterations is equaled M, and ε is stop criterium for k-means iterations.
8. Map $C_1,\ C_2,\ \ldots,\ C_k$ to $S_1, S_2, \ldots, S_k$ such that
$$(\forall j \in \{1, 2, \ldots, k\})(\forall \mathbf{x}_i \in C_j)(\mathbf{x}_i \mapsto p_i \text{ and } p_i \text{ assign to } S_j)$$

Let's pay attention to eigenvectors. The algorithm uses eigenvectors, which are computed by general eigenproblem of matrix $\mathbf{L}$. Due to **Proposition 3**, they are correspond to the graph Laplacian $\mathbf{L}_{rw}$ eigenvectors. Therefore, in fact, the algorithm uses eigenvectors of *normalized Laplacian* $\mathbf{L}_{rw}$ and, hence, it is called *normalized spectral clustering*.
The following spectral clustering algorithm is also *normalized*, but $\mathbf{L}_{sym}$ is used instead of $\mathbf{L}$, respectively $\mathbf{L}_{rw}$, which results that an additional normalized step is needed.

Algorithm 6: Normalized image spectral clustering (Ng, Jordan, Weiss)

Input : digital image Im, detailness parameter $\sigma \in R$, desired region count k, reasonable small $\varepsilon \in \mathbb{R}^+$, maximum iteration count M
Output: regions $S_1, S_2, \dots S_k$

1. By row, stepwise denote Im pixels as the $p_1,\ p_2,\ \dots,\ p_n$, where n is equaled Im resolution, and map them to a vertex set V, see (4.5).
2. Construct an image similarity grid-graph $G = (V, E)$, and use σ as a detailness parameter for the *Gaussian kernel similarity function*. Let $\mathbf{W}$ be its *adjacency matrix*, $\mathbf{D}$ be its *degree matrix*.
3. Determine the *unormalized graph Laplacian* $\mathbf{L}_{sym} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$.
4. Compute the first k eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_k$ of *unormalized graph Laplacian* $\mathbf{L}$.
5. Let $\mathbf{X} \in \mathbb{R}^{n \times k}$ be matrix, which contains the vectors $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_k$ as columns.
6. Form matrix $\mathbf{T} \in \mathbb{R}^{n \times k}$ from $\mathbf{X}$ by normalizing the rows to norm 1.
7. Remap Im pixels $p_1,\ p_2,\ \dots,\ p_n$ to $\mathbb{R}^k$ such that

   $$p_1 \mapsto \mathbf{t}_1,\ p_2 \mapsto \mathbf{t}_2,\ \dots,\ p_n \mapsto \mathbf{t}_n$$

   where $\mathbf{t}_i$ is i-th rows of matrix $\mathbf{T}$.
8. By one of the *Lloyd type clustering algorithms*, cluster points $\mathbf{t}_1,\ \mathbf{t}_2,\ \dots,\ \mathbf{t}_n$ to regions $C_1,\ C_2,\ \dots,\ C_k$. The Lloyd type algorithm maximum of iterations is equaled M, and ε is stop criterium for k-means iterations.
9. Map $C_1,\ C_2,\ \dots,\ C_k$ to $S_1, S_2, \dots, S_k$ such that

   $$(\forall j \in \{1, 2, \dots, k\})(\forall \mathbf{t}_i \in C_j)(\mathbf{t}_i \mapsto p_i \text{ and } p_i \text{ assign to } S_j)$$

In literature, the approach, in which data are modelled as similarity graphs, is often called the *problem of dimensionality reduction* [63], and an appropriate matrix $\mathbf{X}$ or $\mathbf{T}$ are denoted as *eigenmaps*. In image processing field, the eigenmaps play important roles, inasmuch as, they map pixels to *Euclidian linearity separable space* $\mathbb{R}^k$.
Therefore, detection of complicated shapes, e.g. triquetra [64], part of non-trivial objects areas, is reformulated as the linearly separable problem, which can be processed by Lloyd type algorithms or *SVM* [65]. However, there are significant obstacles in practical implementations as we show in a following text, e.g. by decreasing detailness parameter σ, the condition numbers of *Laplacians* are increasing.

4.3.2 Graph cut point of view

As we say at beginning of this chapter, spectral clustering can be interpreted as the optimization of a minimal-cut/max-flow problem, i.e. it is considered to find partitions of a similarity graph G as follows:

- edges within partition have reasonably high weights, i.e input points, in our case pixels, $p_1,\ p_2,\ \ldots,\ p_l$, which are mapped to graph vertices $v_1,\ v_2,\ \ldots,\ v_l$, are similar,
- edges among different groups have reasonably low weights, i.e. points, pixels respectively, are reasonable dissimilar among partitions.

As it is presented in our standard reference [56, p. 9], let G be an image similarity graph with adjacency matrix $\mathbf{W}$. Further, let $A_1 \subset V,\ A_2 \subset V,\ \ldots,\ A_k \subset V$ be graph partitions and $\mathbb{I}_1,\ \mathbb{I}_2, \ldots,\ \mathbb{I}_k$ be stepwise index sets of point indices in appropriate graph partitions. Now, we define $\widetilde{W}$ as follows:

$$\widetilde{W}\left(A_i, A_j\right) := \sum_{k \in \mathbb{I}_i,\ l \in \mathbb{I}_j} w_{kl}. \tag{4.14}$$

Let k be a partition count. For each partition $A_1 \subset V,\ A_2 \subset V,\ \ldots,\ A_k \subset V$ stepwise define its complement, i.e. $\overline{A_1} \subset V,\ \overline{A_2} \subset V,\ \ldots\ \overline{A_k} \subset V$ and appropriate index sets $\overline{\mathbb{I}_1},\ \overline{\mathbb{I}_2},\ \ldots,\ \overline{\mathbb{I}_k}$. Follow, we can formulate the minimum-cut problem as choosing partitions $A_1,\ A_2,\ \ldots,\ A_k$, which minimize objective function

$$\operatorname{cut}\left(A_1,\ A_2,\ \ldots,\ A_k\right) := \frac{1}{2} \sum_{i=1}^{k} \widetilde{W}\left(A_i, \overline{A_i}\right). \tag{4.15}$$

Let's notice that factor $\frac{1}{2}$ avoids to counting each edge twice in the cut. If problem looks easy, this approach does not lead to satisfactory partitions in practical applications. In many cases, the solution separates one individual vertex from the rest of a graph.

As we define in Chapter 2, achieved object regions should be reasonably large, and we desire the same requirement for graph partitions. In literature, there are two most common used objective functions, which meet our requirement, *Hagen and Kahng* introduced RadioCut [54], and *Shi and Malik* presented Ncut [47], and they are defined such that:

$$\operatorname{RadioCut}(A_1,\ A_2,\ \ldots,\ A_k) := \frac{1}{2}\sum_{i=1}^{k}\frac{\widetilde{W}\left(A_i, \overline{A_i}\right)}{|A_i|} = \sum_{i=1}^{k}\frac{\operatorname{cut}\left(A_i, \overline{A_i}\right)}{|A_i|}, \tag{4.16}$$

$$\operatorname{Ncut}(A_1,\ A_2,\ \ldots,\ A_k) := \frac{1}{2}\sum_{i=1}^{k}\frac{\widetilde{W}\left(A_i, \overline{A_i}\right)}{\operatorname{vol}(A_i)} = \sum_{i=1}^{k}\frac{\operatorname{cut}\left(A_i, \overline{A_i}\right)}{\operatorname{vol}(A_i)}. \tag{4.17}$$

In our standard reference [56, p. 10], there RadioCut relaxation for case $k = 2$ is introduced. Inasmuch as RadioCut relaxation for abritrary $k \in \mathbb{N}$ is generalization of this approach, we directly focus on the general version of RadioCut relaxation [56, p. 11].

Let $A_1 \subset V,\ A_2 \subset V,\ \ldots,\ A_k \subset V$ be partitions of a vertex set similarity graph $G = (V, E)$. Further, we define k indicator vectors $\mathbf{h}_j = \left(h_1^j,\ h_2^j,\ \ldots,\ h_n^j\right)^T$, where $n := |V|$, such that

$$\forall i \in \{1,\ 2,\ \ldots,\ n\}\ \mathbf{h}_i^j = \begin{cases} \frac{1}{\sqrt{|A_j|}} & \text{if } v_i \in A_j, \\ 0 & \text{otherwise.} \end{cases} \tag{4.18}$$

Let's form a matrix $\mathbf{H} \in \mathbb{R}^{n\times k}$ as a matrix containing these k vectors in columns. It is obvious that columns $\mathbf{h}_1,\ \mathbf{h}_2,\ \ldots,\ \mathbf{h}_k$ are orthonormal to each other [56, p. 10-11], i.e. $\mathbf{H}^T\mathbf{H} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. By the successive modifications of quadratic form $\mathbf{h}_j^T\mathbf{L}\mathbf{h}_j$, see [56, p. 10], we get equality

$$\mathbf{h}_j^T\mathbf{L}\mathbf{h}_j = \frac{\operatorname{cut}\left(A_i, \overline{A_i}\right)}{|A_i|}, \tag{4.19}$$

moreover,

$$\mathbf{h}_j^T\mathbf{L}\mathbf{h}_j = \left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right)_{jj}, \tag{4.20}$$

i.e. j-th item of a main diagonal $\mathbf{H}^T\mathbf{L}\mathbf{H}$.

Equations (4.16), (4.19), and (4.20) imply the following equalities

$$\operatorname{RadioCut}(A_1,\ A_2,\ \ldots,\ A_k) = \sum_{j=1}^{k}\mathbf{h}_j^T\mathbf{L}\mathbf{h}_j = \sum_{j=1}^{k}\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right)_{jj} = \operatorname{Tr}\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right), \tag{4.21}$$

where Tr denotes a *trace of a matrix*.

Now, we can quite simply define RadioCut relaxed problem

$$\min_{\mathbf{H}\in\mathbb{R}^{n\times k}} \operatorname{Tr}\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right) \quad \text{subject to } \mathbf{H}^T\mathbf{H} = \mathbf{I}. \tag{4.22}$$

(4.22) is the standard trace minimization problem, which follows the *Rayleigh-Ritz theorem* [66, pp. 64 - 69], i.e. solution is given by choosing $\mathbf{H}$ as the matrix, which contains the first k eigenvectors of $\mathbf{L}$ as its columns. In the fact, it corresponds to the *unnormalized spectral clustering algorithm*, which is described in a previous section.

Deriving Ncut relaxation is quite similar to a previous approach. For the first non-trivial case $k = 2$ component count, we refer to [56, p. 12]. As similar as above, we focus on a general Ncut relaxation for $k \in \mathbb{N}$. we define k indicator vectors $\mathbf{h}_j = \left(h_1^j,\ h_2^j,\ \ldots\ h_n^j\right)^T$, where $n := |V|$, such that

$$\forall i \in \{1,\ 2,\ \ldots,\ n\}\ \mathbf{h}_i^j = \begin{cases} \frac{1}{\sqrt{\operatorname{vol}(A_j)}} & \text{if } v_i \in A_j, \\ 0 & \text{otherwise.} \end{cases} \tag{4.23}$$

Next, let's form a matrix $\mathbf{H} \in \mathbb{R}^{n\times k}$ as a matrix containing these k vectors in columns and we can observe that

$$\mathbf{H}^T\mathbf{H} = \mathbf{I}, \tag{4.24}$$

$$\mathbf{h}_j^T\mathbf{D}\mathbf{h}_j = 1, \tag{4.25}$$

and

$$\mathbf{h}_j^T\mathbf{L}\mathbf{h}_j = \frac{\operatorname{cut}\left(A_i, \overline{A_i}\right)}{\operatorname{vol}\left(A_i\right)}. \tag{4.26}$$

For further information about the approach metioned above, we refer to our standard reference [56, p. 13].

Now, we can simply write NCut minimization problem such that

$$\min_{A_1,\ A_2,\ \ldots,A_k} \operatorname{Tr}\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right) \quad \text{subject to } \mathbf{H}^T\mathbf{D}\mathbf{H} = \mathbf{I}. \tag{4.27}$$

Let's relax problem (4.27) by allowing arbitrary real values in a matrix $\mathbf{H}$ and substitute

$$\widetilde{\mathbf{T}} = \mathbf{D}^{\frac{1}{2}}\mathbf{H},$$

then we obtain the following optimization problem

$$\min_{\mathbf{H}\in\mathbb{R}^{n\times k}} \operatorname{Tr}\left(\left(\mathbf{D}^{-\frac{1}{2}}\widetilde{\mathbf{T}}\right)^T\mathbf{L}\mathbf{D}^{-\frac{1}{2}}\widetilde{\mathbf{T}}\right) \quad \text{subject to } \left(\mathbf{D}^{-\frac{1}{2}}\widetilde{\mathbf{T}}\right)^T\mathbf{D}^{-\frac{1}{2}}\widetilde{\mathbf{T}} = \mathbf{I}. \tag{4.28}$$

By successive modifications, a problem formulation (4.28) becomes

$$\min_{H\in\mathbb{R}^{n\times k}} \mathrm{Tr}\left(\widetilde{\mathbf{T}}^T\mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}\widetilde{\mathbf{T}}\right) \text{ subject to } \widetilde{\mathbf{T}}^T\widetilde{\mathbf{T}} = \mathbf{I}. \tag{4.29}$$

It is obvious because it includes $\left(\mathbf{D}^{-\frac{1}{2}}\right)^T = \mathbf{D}^{-\frac{1}{2}}$, $\left(\mathbf{D}^{-\frac{1}{2}}\widetilde{\mathbf{T}}\right)^T = \widetilde{\mathbf{T}}^T\mathbf{D}^{-\frac{1}{2}}$ and $\mathbf{D}^{-\frac{1}{2}}\mathbf{D}^{-\frac{1}{2}} = I$ by standard linear algebra approaches.

As similar as in the formulation of relaxed RadioCut problem, (4.29) is the standard trace minimization problem, which is solved by the matrix $\widetilde{\mathbf{T}}$ containing the first k eigenvectors of the matrix $\mathbf{L}_{sym}$ as its columns. Let's re-substitute $\mathbf{H} = \mathbf{D}^{-\frac{1}{2}}\widetilde{\mathbf{T}}$ and apply **Proposition 3**, then $\mathbf{H}$ consists the first k eigenvectors of the matrix $\mathbf{L}_{rw}$ or the first k generalized eigenvectors $\mathbf{L}\mathbf{e} = \lambda\mathbf{D}\mathbf{e}$. Hence, Ncut corresponds to *Shi and Malik normalized spectral clustering* [47].

4.4 Piecewise smooth image partitioning

For now, we describe problem by the graph theory approaches, which are suitable for practical implementation, however, we do not know what we have to treat for image functions, which are the solution of the spectral clustering problem. In [3, p. 2], we show an approach how to reformulate minimization of *Mumford and Shah functional* [67] to minimization of *Laplace–Beltrami operator* [68], which is often called *Laplace–Beltrami equation* in image processing community.

First, let's start by continuous approach, i.e. input image function $g\,(x,y) \in G_{im}$ is continuous and it is defined on Ω. Let $\Omega_{roi} \subset \Omega$ be the *Region Of Interest*, $u\,(x,y)$, which is defined on Ω_{roi}, be the reasonable image partitioning solution, $B := \partial\Omega_{roi}$, i.e the the boundary set, and $\mathrm{int}\,(\Omega_{roi}) := \Omega_{roi} \setminus B$ Futher, we define *Mumford and Shah functional*, such that

$$\min_{u(x,y)} \{F_{ms}\} := \min_{u(x,y)} \left\{ \alpha \int\limits_{\Omega_{roi}} [g(x,y) - u(x,y)]^2 dS \; + \; \beta \int\limits_{\mathrm{int}(\Omega_{roi})} [\; \nabla u(x,y)\;]^2 \; dS + \gamma|B| \right\}, \tag{4.30}$$

where $\alpha, \beta, \gamma \in \mathbb{R}$, and $\nabla u(x,y)$ is a gradient of function $u(x,y)$, i.e. $\nabla u := \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}$.

Let's consider that $u\,(x,y)$ is optimal, therefore terms $\alpha \int\limits_{\Omega_{roi}} [g(x,y) - u(x,y)]^2 dS$ and $\gamma|B|$ are optimal too. Thus, we give the following minimization problem

$$\min_{u(x,y)} \left\{\widetilde{F_{ms}}\right\} = \min_{u(x,y)} \left\{ \beta \int\limits_{\mathrm{int}(\Omega_{roi})} [\; \nabla u(x,y)\;]^2 \; dS \right\}. \tag{4.31}$$

Let's denote $\rightarrow$ as the limit and observe that $\int\limits_{\text{int}(\Omega_{roi})} \beta \, [\, \nabla u \,]^2 \, dS \rightarrow 0$, when

$$\beta \, [\, \nabla u_{|\text{int}(\Omega_{roi})} \,]^2 \rightarrow 0$$

. Inasmuch as the *Laplace-Beltrami* operator $\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$ is defined as

$$\Delta u = \nabla^2 u = \nabla \cdot \nabla u, \tag{4.32}$$

we can reformulate the problem above as follows: $\beta \Delta u_{|\text{int}(\Omega_{roi})} \rightarrow 0$. Moreover, we consider that solution u is optimal, then we can easily write

$$\Delta u_{|\text{int}(\Omega_{roi})} = 0. \tag{4.33}$$

And now, let's modify right side of equation above such that

$$\Delta u_{|\text{int}(\Omega_{roi})} = \lambda_0 u_0 \tag{4.34}$$

where $\lambda_0 = 0$ and u_0 is a constant function in general. Further, let's extend domain to boundary in (4.34), and generalize (4.34) such that

$$\Delta u = \lambda u. \tag{4.35}$$

The (4.35) is called the *Helmholtz equation*, which is also known as the *Laplacian eigenvalue problem*. This naming is obvious inasmuch as discrete analogy of the *Laplace-Beltrami operator* is the *graph Laplacian* $\mathbf{L}$, we refer to [56, p. 28], [60], and [69, p. 34] for further information.

Let Ω_{roi} be regular area, i.e. area with the *Lipschitz boundary*. Further, let's transform a local similarity w_{cd} in the *graph Laplacian* $\mathbf{L}$ to a distance d_{cd} by the relationship $w_{cd} = \frac{1}{d_{cd}^2}$. Then,

$$(\forall c, d \in \mathbb{I}) \, (d_{cd} \rightarrow 0) \Rightarrow \mathbf{L} \rightarrow \Delta, \tag{4.36}$$

which implies

$$\mathbf{e}_{|\Omega_{roi}} \rightarrow u_{|\Omega_{roi}} \Rightarrow \mathbf{L}\mathbf{e}_{|\Omega_{roi}} \rightarrow \Delta u_{|\Omega_{roi}}, \tag{4.37}$$

where $\mathbf{e}_{|\Omega_{roi}}$ is an eigenvector of $\mathbf{L}$ and it is defined on discretized Ω_{roi}. Therefore, we can consider $\mathbf{e}_{|\Omega_{roi}}$ as the discrete analogy of $u_{|\Omega_{roi}}$. It finally implies

$$\mathbf{L}\mathbf{e}_{|\Omega_{roi}} = \lambda \mathbf{e}_{|\Omega_{roi}} \rightarrow \Delta u_{|\Omega_{roi}} = \lambda u_{|\Omega_{roi}}. \tag{4.38}$$

Thus, *standard Laplacian eigenproblem* is a discrete analogy of solving the *Helmholtz equation*. For $\mathbf{L}_{sym}$, $\mathbf{L}_{rw}$ continuous analogies and their appropriate eigenproblem continuous analogies, we refer to [70, p. 1333].

Now, we focus on the solution of the *Helmholtz equation* for $\lambda = 0$, we consider that $\Omega_{roi} \subset \Omega$ is regular. In case of the Neumann boudary condition $\frac{\partial u}{\partial n} = 0$ and for closed areas without boundary, the solutions of the *Helmholtz equation* are constant functions $u \in L^2$ or $u_{|\text{int}(\Omega_{roi})} \in L^2$, see [60, p. 3] for further information. For $\mathbf{L}$, which is defined on a discretized regular area Ω_{roi}, it implies: *If multiplicity of zero eigenvalue is equaled* 1*, then eigenvector* $\mathbf{e}$ *is constant on discretized* Ω_{roi} *or discretized int* (Ω_{roi}). Hence, *unnormalized spectral clustering* is discrete analogy for minimization problem (4.31). Moreover, we can consider a consequence above as another explanation of **Proposition 1**.

Finally, let's consider multiplicity $k > 1$ of zero eigenvalue of matrix $\mathbf{L}$. From the *standard spectral clustering theory*, we know that eigenvector $\mathbf{e}_j$, which corresponds to zero eigenvalue λ_0^j, has constant entries $(\mathbf{e}_j(v_i) = \alpha, \alpha \in \mathbb{C} \setminus \{0\})\,(\forall v_i \in A_j)$, otherwise is equaled 0. For now, let's re-callA_j as the subset of discretized Ω_{roi} and re-call $\mathbf{e}_j := \frac{1}{\alpha}\mathbf{e}_j$ as the typical representant from Ξ_{A_j}. Without loss of generality, reorder eigenvector $\mathbf{e}_j$ such that

$$\overline{\mathbf{e}}_1^j = \overline{\mathbf{e}}_2^j = \cdots = \overline{\mathbf{e}}_l^j = \alpha \neq \overline{\mathbf{e}}_{l+1}^j, \tag{4.39}$$

$$\overline{\mathbf{e}}_{l+1}^j = \overline{\mathbf{e}}_{l+2}^j = \cdots = \overline{\mathbf{e}}_n^j = 0, \tag{4.40}$$

and remap index set $\mathbb{I} \mapsto \bar{\mathbb{I}}$ as follows: $(\forall i \in \mathbb{I})\left(\nu_i = \bar{\mathbb{I}}(i), \mu_i = \mathbb{I}(\nu_i)\right) : v_{\nu_i} = v_{\mu_i}$ in a graph topology sense.

Let's split $\mathbf{e}_j$ into two vectors as follows

$$\mathbf{e}_j^1 := \left(\overline{\mathbf{e}}_1^j,\ \overline{\mathbf{e}}_2^j,\ \ldots,\ \overline{\mathbf{e}}_l^j\right), \tag{4.41}$$

$$\mathbf{e}_j^2 := \left(\overline{\mathbf{e}}_{l+1}^j, \overline{\mathbf{e}}_{l+2}^j, \ldots, \overline{\mathbf{e}}_n^j\right). \tag{4.42}$$

Therefore, $\mathbf{e}_j^1$ is a solution of the eigenproblem $\mathbf{L}_{\hat{\mathbb{A}}_j}\mathbf{e}_j^1 = \lambda_0^j\mathbf{e}_j^1$, where

$$\hat{\mathbb{A}}_j := \{v_{\mathbb{I}(1)},\ v_{\mathbb{I}(2)},\ \ldots,\ v_{\mathbb{I}(l)}\},$$

$\mathbf{L}_{\hat{\mathbb{A}}_j}$ is *Laplacian*, which is defined on the set $\hat{\mathbb{A}}_j$, and $\mathbf{e}_j^1$ is eigenvector, which corresponds to $\lambda_0^j = 0$ eigenvalue. Let $\Omega_{roi}^1 \subset \Omega_{roi},\ \Omega_{roi}^2 \subset \Omega_{roi},\ \ldots,\ \Omega_{roi}^j \subset \Omega_{roi},\ \ldots,\ \Omega_{roi}^k \subset \Omega_{roi}$ be closed sets without boundaries. Let's assume A_j as discretized Ω_{roi}^j and $\mathbf{e}_j^1$ as the discrete analogy $u_{|\Omega_{roi}^j}$, then

$$\mathbf{L}_{\hat{\mathbb{A}}_j}\mathbf{e}_j^1 \ \rightarrow\ \Delta u_{|\Omega_{roi}^j} \tag{4.43}$$

on regular Ω_{roi}^j and its discrete analogy $\hat{\mathbb{A}}_j$. Inasmuch as $\mathbf{e} \equiv \mathbf{o}$ is not considered as any eigenvector, $\mathbf{e}_j^2$ can not be a solution of eigenproblem $\mathbf{L}_{\overline{\mathbb{A}}_j}\mathbf{e}_j^2 = \lambda_0^j\mathbf{e}_j^2$, where $\overline{\mathbb{A}}_j := V \setminus \hat{\mathbb{A}}_j$, $\mathbf{L}_{\overline{\mathbb{A}}_j}$ is the *Laplacian*, which is defined on the set $\overline{\mathbb{A}}_j$, and $\mathbf{e}_j^2$ is the eigenvector, which corresponds to $\lambda_0^j = 0$ eigenvalue.

Let's re-call $\mathbf{e}_{|\Omega^j_{roi}}$ as the dicrete analogy of $u_{|\Omega^j_{roi}}$, and $\mathbf{L}^j\mathbf{e}_{|\Omega^j_{roi}}$ the dicrete analogy of $\Delta u_{|\Omega^j_{roi}}$, then the propositions above imply

$$\mathbf{L}^j\mathbf{e}_{|\Omega^j_{roi}} = \lambda\mathbf{e}_{|\Omega^j_{roi}} \rightarrow \Delta u_{|\Omega^j_{roi}} = \lambda u_{|\Omega^j_{roi}}. \tag{4.44}$$

Following, let's re-call $\hat{\mathrm{A}}_1$ as discretized Ω^1_{roi}, $\hat{\mathrm{A}}_2$ as discretized Ω^2_{roi}, $\ldots$, $\hat{\mathrm{A}}_k$ as discretized Ω^k_{roi}, and stepwise apply the approach above on $\hat{\mathrm{A}}_1,\ \hat{\mathrm{A}}_2,\ \ldots,\ \hat{\mathrm{A}}_k$, then

$$\mathbf{L}_{\hat{\mathrm{A}}_1}\hat{\mathrm{e}}^1_1 \ \rightarrow\ \Delta u_{|\Omega^1_{roi}},\ \mathbf{L}_{\hat{\mathrm{A}}_2}\hat{\mathrm{e}}^1_2 \ \rightarrow\ \Delta u_{|\Omega^2_{roi}},\ \ldots,\ \mathbf{L}_{\hat{\mathrm{A}}_k}\hat{\mathrm{e}}^1_k \ \rightarrow\ \Delta u_{|\Omega^k_{roi}}. \tag{4.45}$$

Further, let's re-call $\mathbf{e}_{|\Omega^1_{roi}}$ as the dicrete analogy of $u_{|\Omega^1_{roi}}$, $\mathbf{e}_{|\Omega^2_{roi}}$ as the dicrete analogy of $u_{|\Omega^2_{roi}}$, $\ldots$, $\mathbf{e}_{|\Omega^k_{roi}}$ as the dicrete analogy of $u_{|\Omega^k_{roi}}$, and $\mathbf{L}^1\mathbf{e}_{|\Omega^1_{roi}}$ the dicrete analogy of $\Delta u_{|\Omega^1_{roi}}$, $\mathbf{L}^2\mathbf{e}_{|\Omega^2_{roi}}$ the dicrete analogy of $\Delta u_{|\Omega^2_{roi}}$, $\ldots$, $\mathbf{L}^k\mathbf{e}_{|\Omega^k_{roi}}$ the dicrete analogy of $\Delta u_{|\Omega^k_{roi}}$. Then

$$\mathbf{L}^1\mathbf{e}_{|\Omega^1_{roi}} = \lambda\mathbf{e}_{|\Omega^1_{roi}} \rightarrow \Delta u_{|\Omega^1_{roi}} = \lambda u_{|\Omega^1_{roi}}, \tag{4.46}$$

$$\mathbf{L}^2\mathbf{e}_{|\Omega^2_{roi}} = \lambda\mathbf{e}_{|\Omega^2_{roi}} \rightarrow \Delta u_{|\Omega^2_{roi}} = \lambda u_{|\Omega^2_{roi}}, \tag{4.47}$$

$$\vdots$$

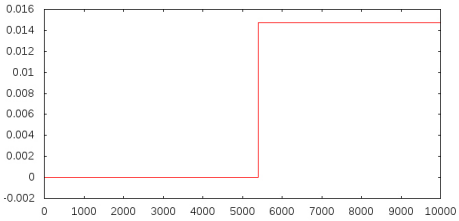
$$\mathbf{L}^k\mathbf{e}_{|\Omega^k_{roi}} = \lambda\mathbf{e}_{|\Omega^k_{roi}} \rightarrow \Delta u_{|\Omega^k_{roi}} = \lambda u_{|\Omega^k_{roi}}. \tag{4.48}$$

Conclusions The equations, proposition and assumption above imply: *if the local similarities w_{cd} are transformed to the distances d_{cd} by the relationship $w_{cd} = \frac{1}{d^2_{cd}}$ and the discretization step goes to zero, then the discrete regions $\hat{\mathrm{A}}_1,\ \hat{\mathrm{A}}_2,\ \ldots,\ \hat{\mathrm{A}}_k$, i.e. graph partitions, are becoming regular areas $\Omega^1_{roi},\ \Omega^1_{roi},\ldots,\Omega^k_{roi}$, see* ***Figure 4.5****, and constant L^2 functions are solutions, which are defined on these areas. Further, the explanation "how is it possible that image-graph splits into partitions" is still remaining. It is very simple, when local similarities go to 0, i.e. the appropriate image pixels are not similar, then distances between these pixels go to $+\infty$. If the set of such pixels is sufficient large, the system, which is represented by graph, splits into the independent subsystems, i.e. the graph partitions, inasmuch as sufficiently distanced systems have not an influence.*

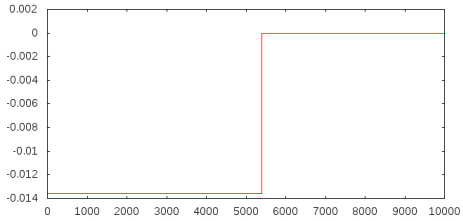
Figure 4.5: Image partitioning by spectral clustering.

At the end of this chapter, we would like to note that the total solution of the Laplacian eigenvectors $\mathbf{e}_1,\ \mathbf{e}_2,\ \ldots,\ \mathbf{e}_k$, corresponds to zero eigenvalues $\lambda_1,\ \lambda_2,\ \ldots,\ \lambda_k$, for each graph-partition are given as the Fourier series $\mathbf{e}_j := \sum_{i=1}^{k} \alpha_i^j \mathbf{e}_i$, where $\alpha_1^j,\ \alpha_2^j,\ \ldots,\ \alpha_k^j$ are the *Fourier series* coefficients. However, a value determinations of these coefficient is beyond the scope of this work and we are going to focus on them in our future works. Typical eigenvector graphs for two connected componets of image-graph, corresponds to **Figure 4.6**, is shown in **Figure 4.7**.

Figure 4.6: 100×100 two-regioned image.



(a) The first partition eigenvector.



(b) The second partition eigenvector.

Figure 4.7: Eigenvectors of **Figure 4.6** appropriate partition.

The *Arnoldi method* [71, pp. 128-134] is typically used as the method for solving spectral clustering eigenproblem and ARPACK[1] library, or PARPACK[2] for parallel implementations, are used as the software eigensolvers, which implement the *Arnoldi method*.

[1] ARPACK can be downloaded from `http://www.caam.rice.edu/software/ARPACK/`.
[2] PARPACK can be downloaded from `http://www.caam.rice.edu/~kristyn/parpack_home.html`

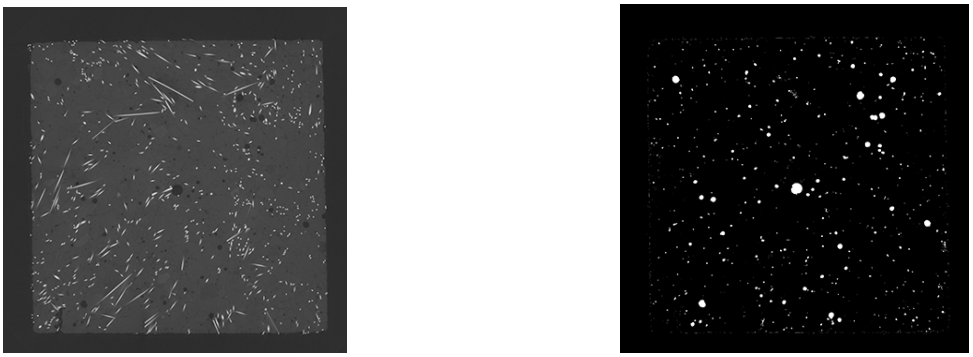
Chapter 5

Image Segmentation Applications in HPC Environtment

In many scientific fields, modern image segmentation methods are often used as an image pre-processing techniques in complex applications ranging from biology and geology to artificial intelligence implementation. Since the end of the last year, we have been developing intensive cooperation with *Institute of Geonics of the CAS* and *University Hospital in Ostrava* in solving of interesting problems, i.e. separation of specific materials in geocomposite CT scans and detecting prostate carcinoma. In this chapter, we present our achieved results and proposed approaches for future work.

5.1 Geological applications

Geocomposite CT scans are typically 8-bit grayscale images; it implies that appropriate intensities of desired materials are decoded into 256 values. Inasmuch as the material areas are not compacts in general, see **Figure 5.1** ,therefore it is suitable to proceed only pixel value distribution and do not to desire their positions in a discrete image area for algorithm-design.



(a) Geocompostite CT scan. (b) Air distribution binary mask.

Figure 5.1: Example of geocompostite CT scan and air distribution in its binary mask.

Thus, we develop two quite new algorithms, i.e. *Simple Method of Reference Materials* and *Method of Reference Materials for Reinforced Ferro Concrete*. These algorithm integral computation parts are based on the k-means algorithm and our algorithm-design loosely follows the techniques, which were introduced in [40]. Moreover, both of our algorithms are suitable for HPC platforms implementation, which is the final goal of this work.

5.1.1 Simple Method of Reference Materials

Simple Method of Reference Materials (SMRM) design-fashion directly follows Steinhaus work [40]. In general, SMRM is the semi-automatized version of the standard Lloyd algorithm. Starting centroids are manually selected by a human segmentator/classifier, then the algorithm continues with the standard Lloyd iteration, i.e. it minimizes the sum of squares within regions with arithmetic mean as the *least-squares* estimator. We prepare the following example in cooperation with *Institute of Geonics of the CAS*. We are very grateful to Dr. Kolcún for consulting and data providing.

For algorithm testing purposes, we develop a desktop program called **PermonGeodecompositer**. It is written in C++ language and based on OpenCV framework [72]. Users can run the program on *Linux*, *Windows* or *OS X* operation systems and easily use it.

By mouse clicking at "Select reference material" window, see **Figure 5.2**, users can select proper reference material values and after that the program automatically runs the k-means algorithm with manually selected initialization of starting centroids. When k-means computation finishes, **PermonGeodecompositer** visualizes achieved results, see **Figure 5.3**.

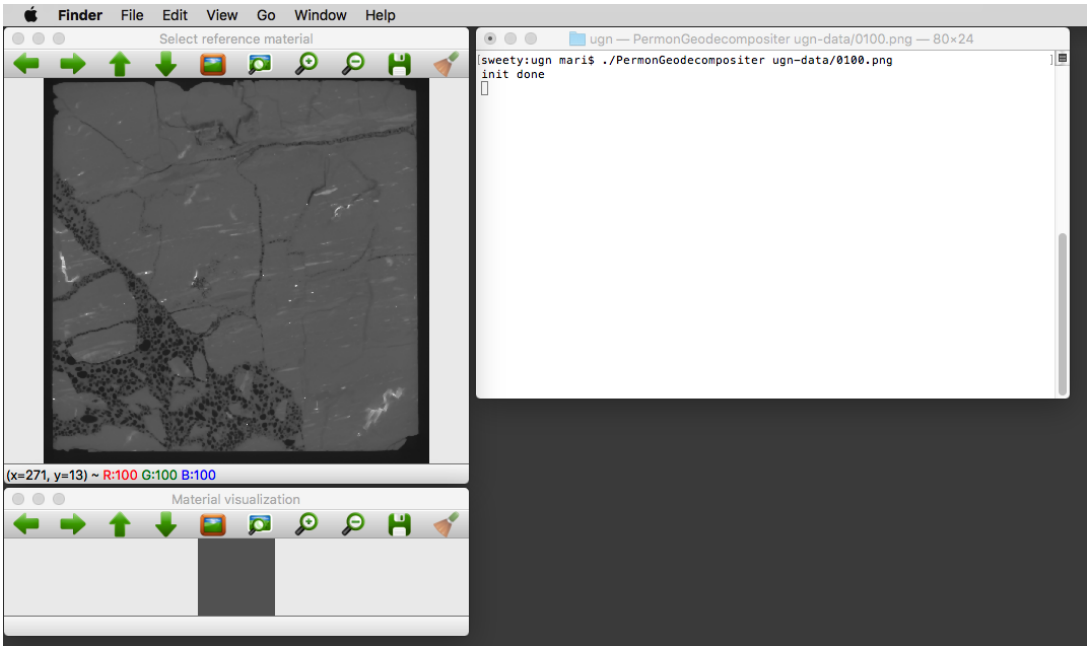


Figure 5.2: PermonGeodecompositer initialization stage.

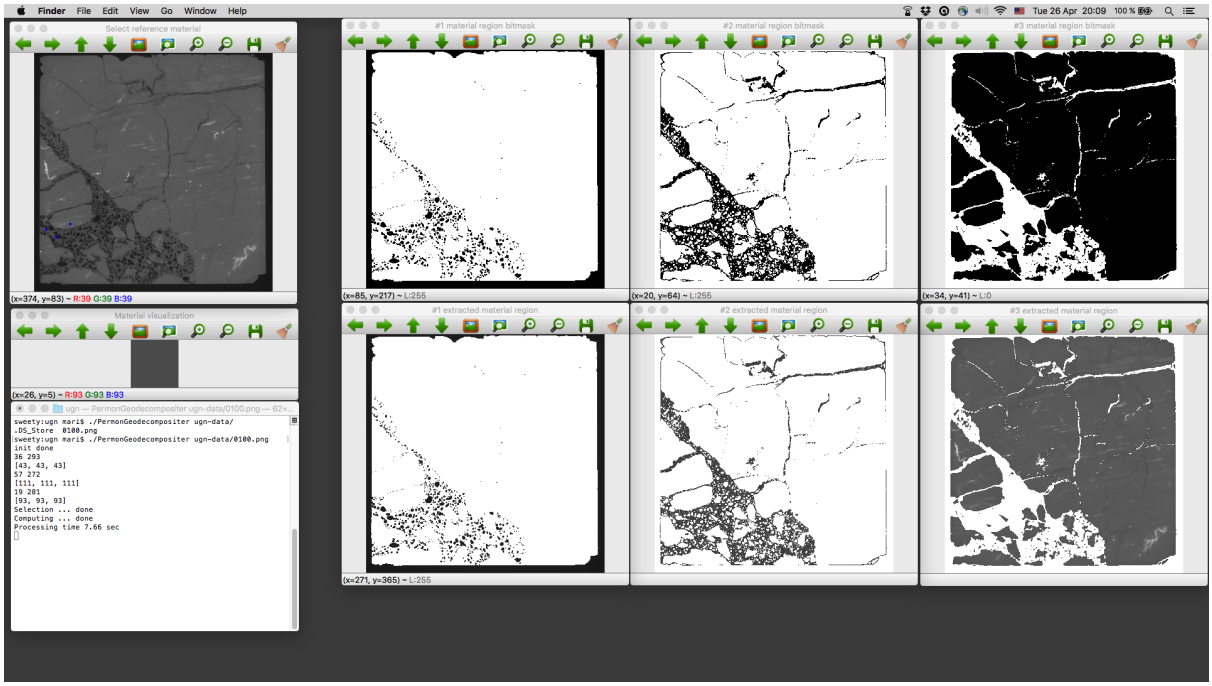


Figure 5.3: PermonGeodecompositer visualization stage.

Detection materials in coal-polyurethane geocomposite

Assignment Separate the regions of three materials, i.e. air, coal, and polyurethane, in a coal-polyurethane geocomposite CT scan, see **Figure 5.4**.

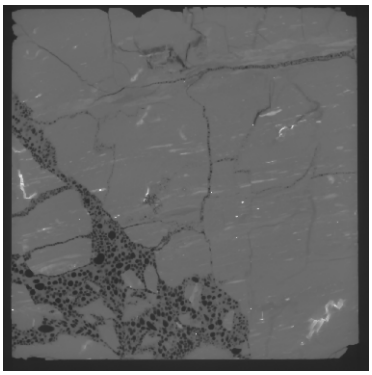


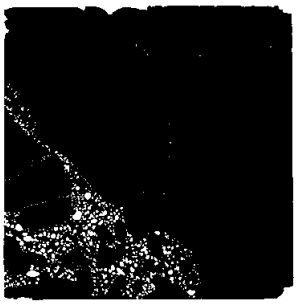
Figure 5.4: Coal-polyurethane geocomposite CT scan.

Solution In **PermonGeodecompositer**, we manually select the proper reference value for each specific material, see **Table 5.1**. Then **PermonGeodecompositer** automatically runs k-means computations.

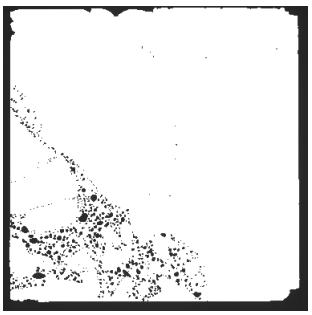
material	value (in grayscale)	color
air	43	
coal	111	
polyurethane	93	

Table 5.1: Manually selected reference values for SMRM algorithm.

Results By the initial conditions for SMRM algorithm mentioned above, we obtain the following results, see **Figure 5.5**, **Figure 5.6**, **Figure 5.7**.

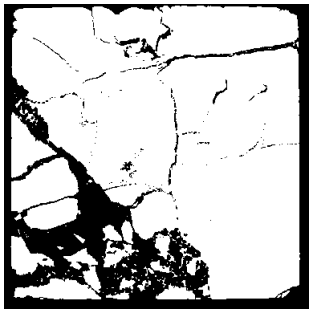


(a) Air region binary mask.

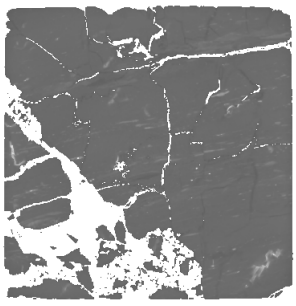


(b) Air extracted region from **Figure 5.4**.

Figure 5.5: Air region achieved result.

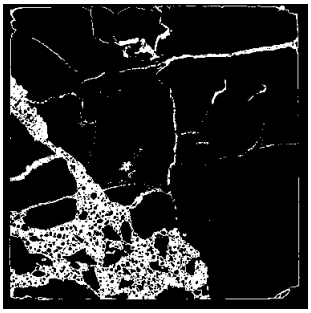


(a) Coal region binary mask.

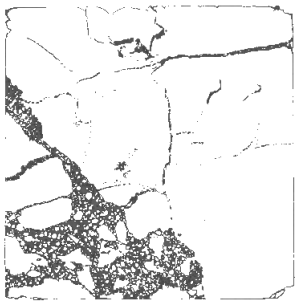


(b) Coal extracted region from **Figure 5.4**.

Figure 5.6: Coal region achieved result.



(a) Polyurethane region binary mask.



(b) Polyurethane extracted region from **Figure 5.4**.

Figure 5.7: Polyurethane region achieved result.

Conclusions In this example, we test SMRM algorithm on one coal-polyurethane CT scan. In our opinion, the algorithm converges to optimal solutions, hence, we propose to use it for detecting the materials on simple geocomposites CT scans.

For testing the presented approach, we use OpenCV parallel implementation of the k-means algorithm, which is based on OpenCL technology. Inasmuch as used implementation of k-means algorithm for our initial conditions converges very quick, an average computation takes about 7.64s on MacBook Pro (late 2012) in CPU-mode and about 0.82s on Salomon Intel Xeon E5-2680v3 CPU, therefore we propose hybrid-parallel implementation for image large dataset as follows: *Run one mpi-process on each allocated processor. MPI master process divides the input data-set among slave-processes. Each mpi-process runs OpenCV k-means on each image from the appropriate input subset. Initial conditions for SMRM algorithm is entered by users.*

Inasmuch as entering initial condition for each image from input data set is prolonged and inefficient, we introduced our automatized algorithm called **Method of Reference Materials** in the next subsection.

5.1.2 Method of Reference Materials for Reinforced Ferro Concrete

We have also prepared another image segmentation application, in cooperation with Dr. Kolcún again. From *Institute of Geonics of the CAS*, we get 1491 CT scans of Reinforced Ferro Concrete (RFC) geocomposite as a gift, 591^{st} image is shown in **Figure 5.8**. Inasmuch as we consider 1491 CT scans for a large data set, we develop an automatized algorithm for detecting air, conctrete and ferro fibre regions. We called this method as the *Method of Reference Material for Reinforced Ferro Concrete* (MRM-RFC).

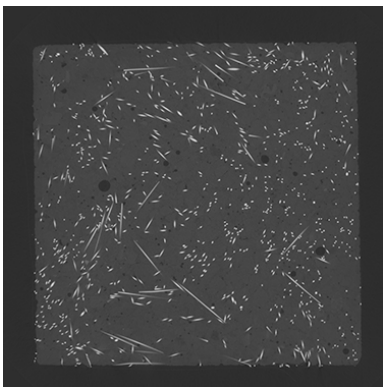


Figure 5.8: 591^{th} image from Reinforced Ferro Concrete data set.

In general, MRM-RFC is designed by the hierarchical image segmentation approach, which we introduce in Chapter 2. The integral computation part is based on modified k-means algorithm, i.e. starting centroids are automatically determined from value distribution and starting centroid values are fixed to the appropriate regions. We call this modification *start-fix k-means*. The hierarchical algorithm approach consists three *mode* iterations as follows:

0^{th} mode iteration algorithm determines the geocomposite *Region of Iterest*, i.e. it removes an outside air region, see **Figure 5.9**, where white color represents *Region of Interest*. We call this *Region of Iterest* as *Problem Region of Iterest*.

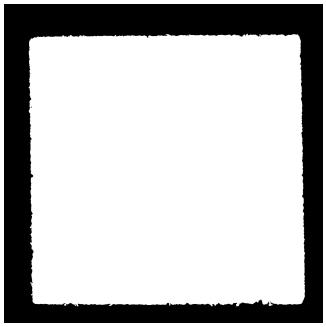
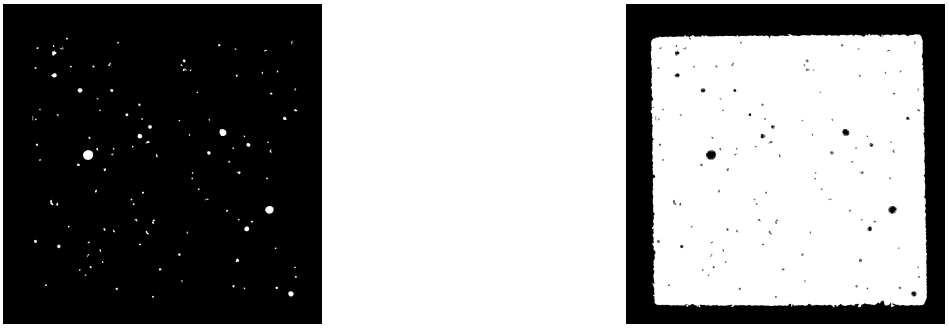


Figure 5.9: MRM-RFC Region of Interest binary mask.

1st mode iteration In the *Problem Region of Iterest* from the previous mode iteration, *MRM-RFC* computes the inner air region, see **Figure 5.10a**, and save it for further processing. Then algorithm removes the inner air region from the *Problem Region of Iterest*, **Figure 5.10b**.



(a) Air bubbles inside geocomposite binary mask. (b) Reinforced Ferro Concrete binary mask.

Figure 5.10: Air bubbles and Reinforced Ferro Concrete binary mask.

2nd mode iteration In this last mode iteration, the algorithm divides the *Problem Region of Iterest* from the previous mode iteration into two parts, i.e. the ferro-fiber region and the region, in which the concrete material is. In this step, it shows that the fuzzy approach is needed instead of hard clustering method for the suitable results achievement. Inasmuch as fuzzy approach is not mentioned in this thesis, we allow ourselves to skip results in this step.

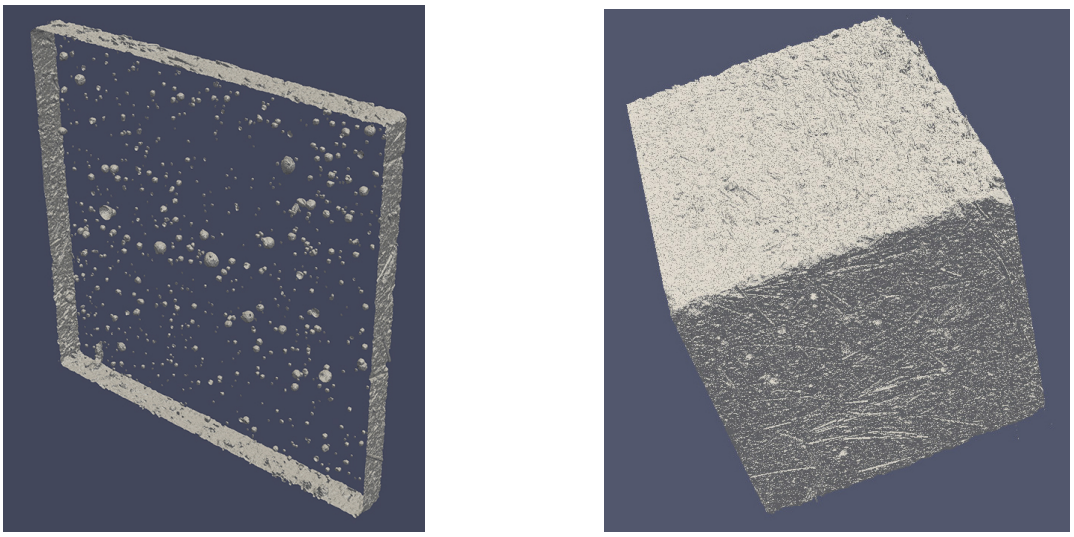
Moreover, we design and implement an augmented verifier, which checks a partial solution after each mode step. If the verification fails, the verifier adjusts an image contrast, shifts initial conditions and runs the k-means again. We implement a procedure for creation 3D mesh of each material region too. Further, we develop complete massive parallel framework for material detection in geocomposite CT scans and creation appropriate 3D meshes. The implementation of the libraries set is based on techniques introduced above. Its core is written in ANSI C and the user interface in C++11. We call our framework **PermonGeoMeshCreator** and we are going to introduce it in our upcoming article [73], in which we focus on proper MRM analysis too. **PermonGeoMeshCreator** framework release is planned in a Fourth Quater 2016.

PermonGeoMeshCreator code examples

```
#include <geo/modules/concrete/rfc/rfc.hpp>

using namespace permon::geo;

int main(int argc, char *argv[]) {
   VERIFICATION_ERROR_TYPE ver_err;
   double fiber_thrs, min_material_shift;

   Image map_roi_air, map_roi_concrete, map_roi_fiber;

   fiber_thrs = 120.;
   min_material_shift = 11.;

   ver_err = PnGeoRFC("rfc_0591.tif", map_roi_air, map_roi_concrete, map_roi_fiber,
                      fiber_thrs, min_material_shift);

   if (!ver_err) {
     PnImageShow(map_roi_air);
     PnImageShow(map_roi_concrete);
     PnImageShow(map_roi_fiber);
   }

   return (0);
}
```

```
#include <geo/mpi/modules/concrete/rfc/rfc.hpp>


using namespace permon::mpi::geo;

int main(int argc, char *argv[]) {
   VERIFICATION_ERROR_TYPE ver_err;
   double fiber_thrs, min_material_shift;

   string in_dir, tmp_dir;
   string output_file_mesh;

   in_dir  = "/scratch/mari/scratch_frc_ugn_11_12/data/";
   tmp_dir = "/scratch/mari/scratch_frc_ugn_11_12/tmp/";

   output_file_mesh = "/scratch/mari/scratch_frc_ugn_11_12/mesh/frc.e";

   fiber_thrs = 120.;
   min_material_shift = 11.;

   ver_err = PnGeoBuildMesh(&argc, &argv, input_dir, tmp_dir, output_file_mesh,
                            wire_thrs, fiber_thrs);

   if (!ver_err) {
     PnMeshShow(output_file_mesh);
   }

   return (0);
}
```

PermonGeoMeshCreator visualization examples



(a) Air distribution in RFC. (b) RFC surface.

Figure 5.11: Reinforced Ferro Concrete visualizations.

5.2 Applications in biomathematics

The cancer incidences rapidly increase. In 2012, there were 14.9 million new cases, or these 7.4 million cases were in men and 6.7 in women, and 8.2 million cancer-related deaths worldwide. Moreover, future prospects are not pink. A new report by the World Health Organization's International Agency for Cancer Research suggests that the incidence of cancer worldwide will have grown by 75% by the year 2030, nearly doubling in some of the developed countries. Statistics are related to [74], [75], [76]. Therefore, we have decided to develop cooperation with *University Hospital in Ostrava* in prostate carcinoma research and tumour detection in pelvis CT scans; typical representant of medical CT scan is shown in **Figure 5.12**.

First, we re-design algorithm for tumour detection based on normalized spectral clustering method, which is introduced in Chapter 4. For data set reduction, the flood algorithm, often called thresholding, is used as follows: *the bones and inner air are separated from a soft tissue by their Hounsfield unit values.* Further, we write the interface to PETSc mathematical library for the numerical approaches on graph structures. This interface is written in ANSI C programming language. After refectoring and final debugging, we are going to present it in our future works.

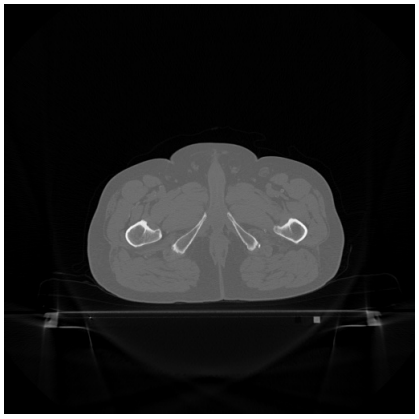
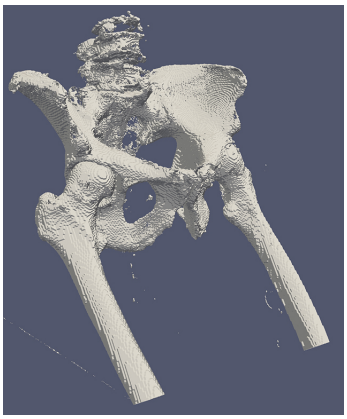
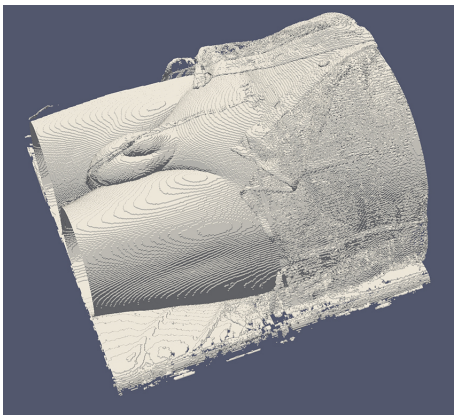


Figure 5.12: Example of pelvis CT scan.

Unfortunately, the problem is occurrs with the *Laplacian* condition number, and it works only on small resolution pictures. Hence, we propose FETI-like approach for the normalized spectral clustering method and we are going to introduce it in our upcoming article [77]. Thererefore, we have prepared other medical applications in cooperation with Mr. Knýbel from *University Hospital in Ostrava*. They are sceleton pelvis part visualization, see **Figure 5.13a**, and body pelvis part visualization, see **Figure 5.13b**; these meshes are generated from 429 DICOM files.



(a) Pelvis skeleton part visualization.



(b) Pelvis body part visualization.

Figure 5.13: Medical visualization examples.

To achieve these solutions, we proceed in a standard way, i.e. the flood algorithm is used for the *Region of Interest* of determination and marching cubes algorithm for mesh generation from *Region of Interest* binary masks. Final meshes are saved into *Exodus II* [78] file format and the Paraview application [79] is used for their visualization. Our algorithm implementation in C++11 programming language is based on the VTK framework.

Chapter 6

Conclusions

> When you can measure what you are speaking about and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of the meager and unsatisfactory kind.
>
> *Lord Kevin*

After two-years of hardworking, we finally reached the check-point, in which we can say: "We are just about finish." But we do not finish at all. In our work, we only show the tip of the iceberg and we still have a long way to go. A few doors are opened in front us, and we have to decide, which way we to go.

Of course, we realize that climbing to the first check point has been really tough. We have been working hard with great effort till the very end. Several times, we have re-labelled the notation, re-written the theory, and debugged our codes. We have been developing quite new image segmentation theory, which is called **Extended Image Segmentation Theory**, and we have exactly proved object topologies in an image scene. Moreover, we have proved the typical region area shape in spectral clustering approaches. Furher, we have designed and implemented two algorithms, i.e. **Simple Method of Reference Materials** and **Method of Reference Materials**, for material detections in geocomposite CT scans and proposed a quite new initialization technique for standard Lloyd clustering algorithm; this technique is called the **regular k-means+2**. Also, a part of our work has been establish cooperation with *University Hospital in Ostrava* on prostate carcinoma research and *Institute of Geonics of the CAS* on material detection in CT scan geocomposites. During these collaborations, we have built a multiplatform desktop application called **PermonGeodecompositer** written in C++, **graph implementation on the top of PETSc library** written in ANSI C, and massively parallel **PermonGeoMeshCreator framework**, which is programmed by combining the ANSI C and C++11 approaches.

Moreover, we submitted [3], and our of two articles are in preparation [73], [77]. Our partial results were presented at **HPCSE 2015**[1], **ICNAAM 2015**[2] and **SC 2015**[3] conferences.

Yes! It has been difficult enough! It has consumed a lot of our free-time! We sacrificed plenty of things! But we have to admit that **it's simply worth it**! And our ride is just beginning ...

[1] `http://industry.it4i.cz/HPCSE2015/`
[2] `http://icnaam.org`
[3] `http://sc15.supercomputing.org`

References

[1] Robert A. Gatenby, Olya Grove, and Robert J. Gillies. Quantitative Imaging in Cancer Evolution and Ecology. *Radiology*, 269(1):8–14, 2013. doi: 10.1148/radiol.13122697. URL http://dx.doi.org/10.1148/radiol.13122697.

[2] FLIR Systems, Inc. http://www.flir.co.uk. Vehicle detection at a level crossing. Digital Image. [Online accessed February 15, 2016]. Available at http://tinyurl.com/zf8x5zl.

[3] Marek Pecha and Martin Čermák. Segmentations for Piecewise Smooth Pictures in PERMON. 2015. Accepted for conference ICNAAM 2015. It will be published in AIP 2016.

[4] James J. DiCarlo, Davide Zoccolan, and Nicole C. Rust. How Does the Brain Solve Visual Object Recognition? *Neuron*, 73(3):415–434, feb 2012. doi: 10.1016/j.neuron.2012.01.010. URL http://dx.doi.org/10.1016/j.neuron.2012.01.010.

[5] Kirt Lillywhite, Dah-Jye Lee, Beau Tippetts, and James Archibald. A feature construction method for general object recognition. *Pattern Recognition*, 46(12):3300–3314, 2013. doi: 10.1016/j.patcog.2013.06.002. URL http://dx.doi.org/10.1016/j.patcog.2013.06.002.

[6] Eric W. Weisstein et al. Image Identification Project. Powered by Wolfram Cloud – Building intelligence into a new kind of cloud. URL https://www.imageidentify.com. WWW. [Online accessed February 18, 2016].

[7] Larry Page, Sergey Brin, et al. Google Self-Driving Car Project. Google Research. URL https://www.google.com/selfdrivingcar/. WWW. [Online accessed February 18, 2016].

[8] Michael Nielsen. http://michaelnielsen.org. Neural Networks and Deep Learning. URL http://neuralnetworksanddeeplearning.com. WWW. [Online accessed February 22, 2016].

[9] Unknown author. Brain icon. Digital Image. [Online accessed February 15, 2016]. Available at http://tinyurl.com/jf9er2n.

[10] Allan Ajifo. Human brain, 2011. Digital Image. Uploaded to www.wikimedia.org. [Online accessed February 15, 2016]. Available at http://tinyurl.com/hn63uge.

[11] Alex David Holub and Greg Griffin. Computational Vision at Caltech. Caltech101 Dataset. URL http://www.vision.caltech.edu/Image_Datasets/Caltech101/. WWW. [Online accessed February 22, 2016].

[12] Li Dajie http://ece.byu.edu/faculty/djlee. Brigham Young University. Eco algorithm sub-region selection feature. Digital Image. [Online accessed February 16, 2016]. Available at http://tinyurl.com/zg4fs8k.

[13] Jacques Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton university bulletin*, 13(49-52):28, 1902.

[14] Saurabh Karsoliya. Approximating number of hidden layer neurons in multiple hidden layer bpnn architecture. *International Journal of Engineering Trends and Technology*, 3(6): 713–717, 2012. ISSN 2231-5381.

[15] Dumitru Ostafe. Neural network hidden layer number determination using pattern recognition techniques. In *2nd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence, SACI*, 2005.

[16] Alan C. Bovik. *Handbook of Image and Video Processing (Communications, Networking and Multimedia)*. Academic press, 2005. ISBN 0-12-119792-1.

[17] Azriel Rosenfeld, Avinash C. Kak, and A. C. Kak. *Digital Picture Processing, Volume 1, Second Edition (Computer Science and Applied Mathematics)*. Morgan Kaufmann, 1982. ISBN 0125973012.

[18] David Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952. doi: 10.1109/jrproc.1952.273898. URL http://dx.doi.org/10.1109/JRPROC.1952.273898.

[19] William B. Pennebaker and Joan L. Mitchell. *JPEG: Still Image Data Compression Standard (Digital Multimedia Standards) (Digital Multimedia Standards S)*. Springer, 1992. ISBN 0442012721. URL http://tinyurl.com/gn658sn.

[20] A. Skodras, C. Christopoulos, and T. Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal Process. Mag.*, 18(5):36–58, 2001. doi: 10.1109/79.952804. URL http://dx.doi.org/10.1109/79.952804.

[21] Ingrid Daubechies. *Ten Lectures on Wavelets*. Society for Industrial & Applied Mathematics (SIAM), 1992. doi: 10.1137/1.9781611970104. URL http://dx.doi.org/10.1137/1.9781611970104.

[22] Dwight Hooker. https://en.wikipedia.org/wiki/Dwight_Hooker. Playboy magazine November 1972. Lena Söderberg. Behind lens, 1972. URL http://tinyurl.com/zj8zzws. Digital image. Uploaded to http://kevinrye.net. [Online accessed March 15, 2016].

[23] Lakshman Prasad and S. Sitharama Iyengar. *Wavelet Analysis with Applications to Image Processing*. CRC Press, 1997. ISBN 978-0849331695. URL http://tinyurl.com/h8ssag2.

[24] David Guy. *Singular sets of minimizers for the Mumford-Shah functional*, volume 233. Springer Science & Business Media, 2006. ISBN 978-3-7643-7302-3. EBook.

[25] Ron Brinkmann. *The Art and Science of Digital Compositing, Second Edition: Techniques for Visual Effects, Animation and Motion Graphics (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann, 2008. ISBN 980-0123706386.

[26] Jakub Šťasta. `http://warag.deviantart.com`. Tatra t603, 2009. Digital Image. [Online accessed September 14, 2015]. Available at `http://fav.me/d2f67ap`.

[27] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986. doi: 10.1109/tpami.1986.4767851. URL `http://dx.doi.org/10.1109/TPAMI.1986.4767851`.

[28] M. H. McAndrew R. L. Adler, A. G. Konheim. Topological entropy. *Transactions of the American Mathematical Society*, 114(2):309–319, 1965. ISSN 00029947. URL `http://www.jstor.org/stable/1994177`.

[29] Lawrence Craig Evans and Ronald F. Gariepy. *Measure Theory and Fine Properties of Functions, Revised Edition (Textbooks in Mathematics)*. Chapman and Hall/CRC, 2015. ISBN 1482242389. URL `http://tinyurl.com/hgp2g4d`.

[30] V.P. Camillo, D. Khurana, T.Y. Lam, W.K. Nicholson, and Y. Zhou. Continuous modules are clean. *Journal of Algebra*, 304(1):94–111, 2006. doi: 10.1016/j.jalgebra.2006.06.032. URL `http://dx.doi.org/10.1016/j.jalgebra.2006.06.032`.

[31] Glenn Meling. `https://plus.google.com/+GlennMelingPhotography`. Båstnäs scrap yard, 2013. URL `http://tinyurl.com/zcfkbmb`. Digital image. [Online accessed March 30, 2016].

[32] Shisong Zhu and Toshio Koga. Feature point tracking for car speed measurement. In *APCCAS 2006 - 2006 IEEE Asia Pacific Conference on Circuits and Systems*. Institute of Electrical & Electronics Engineers (IEEE), 2006. doi: 10.1109/apccas.2006.342343. URL `http://dx.doi.org/10.1109/APCCAS.2006.342343`.

[33] Unknown author. English tudor architectural style, 2012. Digital Image. [Online accessed April 5, 2016]. Available at `http://tinyurl.com/jomyaop`.

[34] S. Sapna Varshney, N. Rajpal, and R. Purwar. Comparative study of image segmentation techniques and object matching using segmentation. In *2009 Proceeding of International Conference on Methods and Models in Computer Science (ICM2CS)*. Institute of Electrical & Electronics Engineers (IEEE), dec 2009. doi: 10.1109/icm2cs.2009.5397985. URL `http://dx.doi.org/10.1109/ICM2CS.2009.5397985`.

[35] Jon Kleinberg. An impossibility theorem for clustering. pages 446–453. MIT Press, 2002.

[36] `https://www.flickr.com/photos/anieto2k/` Andrés Nieto Porras. Dyrhólaey arc. Digital Image. Uploaded to Flicker. [Online accessed April 12, 2016]. Available at `http://tinyurl.com/jer92wm`.

[37] H. S. Black and J. O. Edson. Pulse code modulation. *Trans. Am. Inst. Electr. Eng.*, 66(1):895–899, 1947. doi: 10.1109/t-aiee.1947.5059525. URL http://dx.doi.org/10.1109/T-AIEE.1947.5059525.

[38] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. doi: 10.1109/tit.1982.1056489.

[39] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[40] Hugo Steinhaus. Sur la division des corps matériels en parties. In *Bulletin of the Polish Academy of Sciences*, volume 4, page 801–804. PPolish Academy of Technical Sciences, 1957.

[41] Makoto Matsumoto and Takuji Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, jan 1998. doi: 10.1145/272991.272995. URL http://dx.doi.org/10.1145/272991.272995.

[42] Pixar Animation Studios. Wall • e, 2008. URL http://pixar.wikia.com/wiki/WALL•E. Digital Image. Uploaded to www.wikia.com. [Online accessed April 19, 2016].

[43] Marina Meila and Jianbo Shi. Learning segmentation by random walks. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 873–879. MIT Press, 2001. URL http://papers.nips.cc/paper/1830-learning-segmentation-by-random-walks.pdf.

[44] Miklos Kurucz, Andras Benczur, Karoly Csalogany, and Laszlo Lukacs. Spectral clustering in telephone call graphs. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis - WebKDD/SNA-KDD '07*. Association for Computing Machinery (ACM), 2007. doi: 10.1145/1348549.1348559. URL http://dx.doi.org/10.1145/1348549.1348559.

[45] Scott Doyle, Shannon Agner, Anant Madabhushi, Michael Feldman, and John Tomaszewski. Automated grading of breast cancer histopathology using spectral clusteringwith textural and architectural image features. In *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. Institute of Electrical & Electronics Engineers (IEEE), 2008. doi: 10.1109/isbi.2008.4541041. URL http://dx.doi.org/10.1109/ISBI.2008.4541041.

[46] Motoki Shiga, Ichigaku Takigawa, and Hiroshi Mamitsuka. A spectral clustering approach to optimally combining numericalvectors with a modular network. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*. Association for Computing Machinery (ACM), 2007. doi: 10.1145/1281192.1281262. URL http://dx.doi.org/10.1145/1281192.1281262.

[47] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. doi: 10.1109/34.868688. URL http://dx.doi.org/10.1109/34.868688.

[48] Tao Xiang and Shaogang Gong. Spectral clustering with eigenvector selection. *Pattern Recognition*, 41(3):1012–1029, mar 2008. doi: 10.1016/j.patcog.2007.07.023. URL http://dx.doi.org/10.1016/j.patcog.2007.07.023.

[49] Xiangrong Zhang, Licheng Jiao, Fang Liu, Liefeng Bo, and Maoguo Gong. Spectral clustering ensemble applied to SAR image segmentation. *IEEE Trans. Geosci. Remote Sensing*, 46(7):2126–2136, 2008. doi: 10.1109/tgrs.2008.918647. URL http://dx.doi.org/10.1109/TGRS.2008.918647.

[50] B. Levy. Laplace-beltrami eigenfunctions towards an algorithm that "understands" geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*. Institute of Electrical & Electronics Engineers (IEEE). doi: 10.1109/smi.2006.21. URL http://dx.doi.org/10.1109/SMI.2006.21.

[51] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973. doi: 10.1147/rd.175.0420. URL http://dx.doi.org/10.1147/rd.175.0420.

[52] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23 (2):298–305, 1973.

[53] Alex Pothen, Horst D. Simon, and Kang-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM. J. Matrix Anal. & Appl.*, 11(3):430–452, 1990. doi: 10.1137/0611030. URL http://dx.doi.org/10.1137/0611030.

[54] L. Hagen and A.B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9): 1074–1085, 1992. doi: 10.1109/43.159993. URL http://dx.doi.org/10.1109/43.159993.

[55] Stephen Guattery and Gary L. Miller. On the quality of spectral separators. *SIAM. J. Matrix Anal. & Appl.*, 19(3):701–719, 1998. doi: 10.1137/s0895479896312262. URL http://dx.doi.org/10.1137/S0895479896312262.

[56] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4): 395–416, 2007. doi: 10.1007/s11222-007-9033-z.

[57] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity (Dover Books on Computer Science)*. Dover Publications, 1998. ISBN 0486402584. URL http://tinyurl.com/zu4mlxu.

[58] Eric W. Weisstein. Grid graph, . From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/GridGraph.html.

[59] Fan R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, 1997. ISBN 978-0-8218-0315-8. URL http://tinyurl.com/zyjhrgr.

[60] Martin Reuter, Franz-Erich Wolter, Martha Shenton, and Marc Niethammer. Laplace–beltrami eigenvalues and topological features of eigenfunctions for statistical shape

analysis. *Computer-Aided Design*, 41(10):739–755, oct 2009. doi: 10.1016/j.cad.2009.02.007. URL http://dx.doi.org/10.1016/j.cad.2009.02.007.

[61] Francois Fouss, Alain Pirotte, Jean michel Renders, and Marco Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. Knowl. Data Eng.*, 19(3):355–369, 2007. doi: 10.1109/tkde.2007.46. URL http://dx.doi.org/10.1109/TKDE.2007.46.

[62] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems.*, pages 849–856. MIT Press, 2001.

[63] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. doi: 10.1162/089976603321780317. URL http://dx.doi.org/10.1162/089976603321780317.

[64] Eric W. Weisstein. Triquetra, . From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Triquetra.html.

[65] Xu Yu, Jing Yang, and Jian pei Zhang. A transductive support vector machine algorithm based on spectral clustering. *AASRI Procedia*, 1:384–388, 2012. doi: 10.1016/j.aasri.2012.06.059. URL http://dx.doi.org/10.1016/j.aasri.2012.06.059.

[66] Helmut Lütkepohl and H. Lutkepohl. *Handbook of Matrices*. Wiley, 1997. ISBN 978-0471970158. URL http://tinyurl.com/hwa37ea.

[67] David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, jul 1989. doi: 10.1002/cpa.3160420503.

[68] Guy Rosman, Xue-Cheng Tai, Lorina Dascal, and Ron Kimmel. Polyakov action minimization for efficient color image processing. In *Trends and Topics in Computer Vision*, pages 50–61. Springer Science + Business Media, 2012. doi: 10.1007/978-3-642-35740-4_5. URL http://dx.doi.org/10.1007/978-3-642-35740-4_5.

[69] Max Wardetzky, Saurabh Mathur, Felix Kälberer, and Eitan Grinspun. Discrete laplace operators: No free lunch. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 33–37, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. ISBN 978-3-905673-46-3. URL http://dl.acm.org/citation.cfm?id=1281991.1281995.

[70] Matthias Hein, Jean-Yves Audibert, and Ulrike von Luxburg. Graph laplacians and their convergence on random neighborhood graphs. *J. Mach. Learn. Res.*, 8:1325–1370, December 2007. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1314498.1314544.

[71] Youcef Saad. *Numerical methods for large eigenvalue problems*, volume 158. SIAM, 1992.

[72] G. Bradski. *Dr. Dobb's Journal of Software Tools*, 2000.

[73] Marek Pecha and Martin Čermák. Method of the reference materials for fiber-reinforced concrete. 2016. Article in preperation. It will be submit into special issue of Applied Mathematics and Computation.

[74] Thomas H. Maugh II. World cancer incidence will grow 75% by 2030, who says. URL `http://tinyurl.com/j9nycbz`. WWW. [Online accessed April 28, 2016].

[75] World Cancer Research Fund International. Worldwide data. URL `http://tinyurl.com/o24cnc4`. WWW. [Online accessed April 28, 2016].

[76] National Cancer Institute. Cancer statistics. URL `http://tinyurl.com/j6zpl2q`. WWW. [Online accessed April 28, 2016].

[77] Marek Pecha and Martin Čermák. Feti-like approaches on spectral clustering techniques. 2016. Article in preperation. It will be submit into conference ICNAAM 2016.

[78] Larry A Schoof and Victor R Yarberry. Exodus ii: a finite element data model. Technical report, Sandia National Labs., Albuquerque, NM (United States), 1994.

[79] A. Henderson. Paraview guide, a parallel visualization application, 2007. URL `http://www.kitware.com/products/paraview.html`. Kitware Inc.